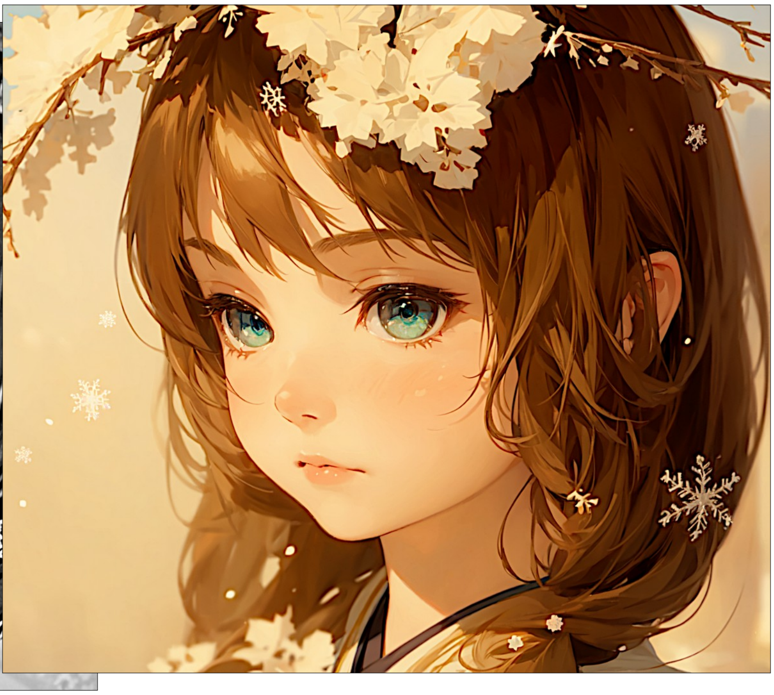
KAZAHANA a.k.a. Snowflowerwind - **The 'Snow Flurry' Searcheress** -
The typhoon class exact & wildcards & fuzzy (Wagner-Fischer) "on the fly" searcheress

An x-gram superfast full-text 'on the fly' suggesteress using exact & wildcards & Levenshtein distance matching
100% FREE, written in C, HEXADECAD-Threaded Linux & Windows executables
Free download (soon, enabling UTF-8 needles) at: www.sanmayce.com/Downloads/Kazahana_UTF8.zip



"Kazahana" (or "kazabana"--same word different reading, probably depending on the locality) is a meteorological phenomenon in which snow falling in a location quite remote from you is blown by strong winds and flutter in the sunny sky above you. Therefore, it is "snow flower," and "wind flower" and "snowing on a clear day." This doesn't happen everywhere but occurs in certain parts of Japan where the conditions are right for the phenomenon to happen. I've seen it in Kyoto and Gunma, but not in Tokyo where I live. Because of the beauty and romance of snowflakes glittering in the sun, Kazahana is popularly used as names of products, stores, etc. That is the reason why you saw it used for a doll.

Диана Експрес - Северина / Diana Express - Severina

Северина, момиче от сняг / Severina a girl made of snow
всяка зима е северен знак / every winter she is a northern sign
аз го имам в песен от юг / I have it in song from south
Северина - радост за друг / Severina - a joy not for me

И като сняг тихо вали / And like snow, she silently comes
вик от мойта любов / a scream from my love
и се топи и навява тъга / and she melts and evokes sadness
песента ми за теб / my song for you

Северина, момиче от сняг / Severina a snow girl
на приказна фея / she is a fabulous fairy's
е северен знак / northern sign
целува ме силно и досега! / she kisses me strongly, still!
тръгва зима - бяла тъга! / winter starts marching on - white sadness!

Mitko Shterev - keyboards
Illya Angelov - lead vocal & guitar
https://www.youtube.com/watch?v=3AIjDY1Gc3c

The heavy benchmark, on Linux:
The file being fuzzed: 'enwiki-20241001-pages-articles.xml' 104,233,480,802 bytes
The needle (Edit Distance 4): "Silvestor Staloune"
Hit lines found by **Kazahana**: 3,568
Hit lines (unique) found by **DraFF**: 19
The testmachine: Laptop ThinkPad P14s Gen 4, CPU: AMD Ryzen 7 PRO 7840U, 8 Cores, RAM: 64GB LPDDR5 Configured Memory Speed: 6400 MT/s

```
+---------------------------------------+---------------+---------------------+
¦ Finder tool                           ¦     Wall Time ¦     CPU utilization ¦
+---------------------------------------+---------------+---------------------+
¦ DraFF.elf                             ¦ 4,610 seconds ¦ 15.855 CPUs utilized ¦
¦ kazahana_clang_18.1.6.x64.dynamic.elf ¦ 7,641 seconds ¦ 14.968 CPUs utilized ¦
¦ kazahana_gcc_14.1.1.x64.dynamic.elf   ¦ 9,947 seconds ¦ 14.511 CPUs utilized ¦
+---------------------------------------+---------------+---------------------+
```

Contents:

CPU benchmark – **Kazahana** fuzzy-proofing Wikipedia for “**Sylvester Stallone**” misspellings

K
A
Z
A
H
A
N
A

An old article: https://www.codeproject.com/Articles/683665/Fastest-Exact-Fuzzy-Wildcard-Full-text-Searcher
Homethread: https://www.linuxquestions.org/questions/programming-9/need-assistance-in-running-16-threaded-superheavy-fuzzy-search-4175529384/page2.html#post6163267
The benchmark thread: https://www.overclock.net/forum/21-benchmarking-software-discussion/1754066-cpu-benchmark-finding-linus-torvalds.html

Machinely yours Kaze (sanmayce@sanmayce.com),
2014-Dec-11, 2025-Jan-06

make_ELF.sh:

```
clang kazahana.c -o kazahana_clang.x64.dynamic.elf -O3 -m64 -msse4.2 -funroll-loops -fopenmp -DCommence_OpenMP -D_FILE_OFFSET_BITS=64 -D_gcc_mumbo_jumbo_ -D_POSIX_ENVIRONMENT_
gcc kazahana.c -o kazahana_gcc.x64.dynamic.elf -O3 -m64 -msse4.2 -funroll-loops -fopenmp -DCommence_OpenMP -D_FILE_OFFSET_BITS=64 -D_gcc_mumbo_jumbo_ -D_POSIX_ENVIRONMENT_
```

runbench.sh:

```
# Run first $ su
# Fuzzy search allowing up to 4 errors:
# 1] substitution: 'y' was misspelled as 'i'
# 2] substitution: 'e' was misspelled as 'o'
# 3] deletion: 'l' was removed
# 4] insertion: 'u' was added
lscpu
dmidecode --type 1
dmidecode --type 17
# Set CPU to performance mode for all CPUs
echo "Setting CPU to performance mode..."
for cpu in /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor; do
    echo performance | tee $cpu > /dev/null
done
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
sh -c 'echo 3 >/proc/sys/vm/drop_caches'
#                                                                 "Sylvester Stallone"
nice -n -20 perf stat -d ./kazahana_clang_18.1.6.x64.dynamic.elf 4e "Silvestor Staloune" enwiki-20241001-pages-articles.xml 458752
sh -c 'echo 3 >/proc/sys/vm/drop_caches'
#                                                               "Sylvester Stallone"
nice -n -20 perf stat -d ./kazahana_gcc_14.1.1.x64.dynamic.elf 4e "Silvestor Staloune" enwiki-20241001-pages-articles.xml 458752
sh -c 'echo 3 >/proc/sys/vm/drop_caches'
nice -n -20 perf stat -d ./DraFF.elf enwiki-20241001-pages-articles.xml "Silvestor Staloune" 4
sort -u enwiki-20241001-pages-articles.xml.hits -o enwiki-20241001-pages-articles.xml.hits.unique
cat enwiki-20241001-pages-articles.xml.hits.unique
```

```
[sanmayce@djudjeto SS_vs_enwiki]$ cat enwiki-20241001-pages-articles.xml.hits.unique
Salvestor Stallone
Silvester Stallone
=Silvester Stalone
Silvester Stalone
Silvestro Stallone
:Silvestr Stallone
Silvestr Stallone]
Sylvester Stallone
 Sylvester Stalone
>Sylvester Stalone
[Sylvester Stalone
|Sylvester Stalone
Sylvester Stalone
Sylvester Stalone,
Sylvester Stalone.
Sylvester Stalone<
Sylvester Stalone]
Sylvester Stalone|
Sylvestor Stallone
[sanmayce@djudjeto SS_vs_enwiki]$
```

```
// gcc -O3 -msse4.2 -o DraFF.elf DraFF.c -fopenmp -D_FILE_OFFSET_BITS=64 -DMaxThreads
// ./DraFF.elf large_file.txt "target_string" 3

// [sanmayce@djudjeto7 ~]$ perf stat -d ./DraFF.elf masakari.wrd "and then" 2 | more
// Searching using 4 threads for pattern 'and then' ...
// Thread 1: Match found at file offset 180928: 'antithen' (Edit Distance: 2)
// Thread 3: Match found at file offset 1260482: 'and  fen' (Edit Distance: 2)
// Thread 1: Match found at file offset 335539: 'and  ben' (Edit Distance: 2)
// Thread 0: Match found at file offset 3443683: 'an  than' (Edit Distance: 2)

/*
________                                            .__  .__
\______ \____________     ____   ____   ____   ____ |  | |  | _____
 |    |  \_  __ \__  \   / ___\ /  _ \ /    \_/ __ \|  | |  | \__  \    ______
 |    `   \  | \// __ \_/ /_/  >  <_> )   |  \  ___/|  |_|  |__/ __ \_ /_____/
/_______  /__|  (____  /\___  / \____/|___|  /\___  >____/____(____  /
        \/           \//_____/             \/     \/               \/

___________                              _________                           .__
\_   _____/_ _____________________.__.  /   _____/ ____ _____ _______   ____ |  |__   ___________
 |    __)|  |  \___   /\___   <   |  |  \_____  \_/ __ \__  \\_  __ \_/ ___\|  |  \_/ __ \_  __ \
 |     \ |  | //    /  /    /  \___  |  /        \  ___/ / __ \|  | \/\  \___|   Y  \  ___/|  | \/
 \___  / |____//_____ \/_____ \/ ____| /_______  /\___  >____  /__|    \___  >___|  /\___  >__|
     \/              \/      \/\/              \/     \/     \/            \/     \/     \/

*/

#define _FILE_OFFSET_BITS 64
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <omp.h>

// Function to calculate the Edit Distance between two strings
int computeEditDistance(const char* str1, const char* str2) {
    size_t len1 = strlen(str1);
    size_t len2 = strlen(str2);

    // Allocate DP table
    int dp[len1 + 1][len2 + 1];

    // Initialize the DP table
    for (size_t i = 0; i <= len1; i++) {
        dp[i][0] = i;
    }
    for (size_t j = 0; j <= len2; j++) {
        dp[0][j] = j;
    }

    // Fill the DP table
    for (size_t i = 1; i <= len1; i++) {
        for (size_t j = 1; j <= len2; j++) {
            int cost = (str1[i - 1] == str2[j - 1]) ? 0 : 1;
            dp[i][j] = dp[i - 1][j - 1] + cost;  // Substitution
            if (dp[i][j - 1] + 1 < dp[i][j]) dp[i][j] = dp[i][j - 1] + 1; // Insertion
            if (dp[i - 1][j] + 1 < dp[i][j]) dp[i][j] = dp[i - 1][j] + 1; // Deletion
        }
    }

    // Return the computed Edit Distance
    return dp[len1][len2];
}

void makePrintable(char* str, size_t len) {
    for (size_t i = 0; i < len; i++) {
        if (str[i] == '\0' || str[i] == '\t' || str[i] == '\r' || str[i] == '\n') {
            str[i] = ' '; // Replace with SPC (ASCII 32)
        }
    }
}

int main(int argc, char* argv[]) {
    if (argc != 4) {
        fprintf(stderr, "Usage: %s <string1_file> <string2> <MinimumEditDistance>\n", argv[0]);
        return EXIT_FAILURE;
    }

//omp_set_num_threads(1);
#ifdef MaxThreads
omp_set_num_threads(omp_get_max_threads());
#endif

    // Parse arguments
    const char* string1_file = argv[1];
    const char* string2 = argv[2];
    int minimumEditDistance = atoi(argv[3]);
    int minimumEditDistanceEXTRA = 0;

    // Generate output filename
    char outputFilename[1024];
```

```
    snprintf(outputFilename, sizeof(outputFilename), "%s.hits", string1_file);

    // Open the output file for writing
    FILE* outputFile = fopen(outputFilename, "w");
    if (!outputFile) {
        perror("Error opening output file");
        return EXIT_FAILURE;
    }

    // Open the file containing string1
    FILE* file = fopen(string1_file, "rb");
    if (!file) {
        perror("Error opening file");
        return EXIT_FAILURE;
    }

    // Determine file size using fseeko and ftello
    fseeko(file, 0, SEEK_END);
    off_t file_size = ftello(file);
    rewind(file);

    // String2 length and chunk overlap
    size_t len2 = strlen(string2);
    size_t chunk_size = 1024 * 1024; // 1 MB chunks for processing
    size_t overlap = len2 + minimumEditDistanceEXTRA - 1;

printf("Searching using %d threads for pattern '%s' ...\n",
                               omp_get_max_threads(), string2);

    // Parallel processing
    #pragma omp parallel
    {
        char* buffer = (char*)malloc(chunk_size + overlap);
        if (!buffer) {
            fprintf(stderr, "Memory allocation failed\n");
            exit(EXIT_FAILURE);
        }

        #pragma omp for schedule(dynamic)
        for (off_t chunk_start = 0; chunk_start < file_size; chunk_start += chunk_size) {
            // Determine how much to read
            size_t to_read = chunk_size;
            if (chunk_start + chunk_size > file_size) {
                to_read = file_size - chunk_start;
            }

            // Read chunk with overlap
            fseeko(file, chunk_start, SEEK_SET);
            fread(buffer, 1, to_read + overlap, file);

            // Process chunk
            for (size_t i = 0; i < to_read; i++) {
                char* sub_str = strndup(&buffer[i], len2 + minimumEditDistanceEXTRA); // ASCII 000 c
                //if (!sub_str) {
                //    fprintf(stderr, "Memory allocation failed\n");
                //    exit(EXIT_FAILURE);
                //}
                                        makePrintable(sub_str, len2 + minimumEditDistanceEXTRA);

                int editDistance = computeEditDistance(sub_str, string2);
                if (editDistance <= minimumEditDistance) {
                    #pragma omp critical
                    {
                        printf("Thread %d: Match found at file offset %ld: '%s' (Edit Distance: %d)\n",
                               omp_get_thread_num(), chunk_start + i, sub_str, editDistance);
                              fprintf(outputFile, "%s\n", sub_str);
                    }
                }

                free(sub_str);
            }
        }

        free(buffer);
    }

    fclose(file);
    fclose(outputFile);

    return EXIT_SUCCESS;
}
```

The complete console log:

```
[sanmayce@djudjeto SS_vs_enwiki]$ ls -l
-rwxrwxrwx. 1 sanmayce sanmayce         6608 Dec 31 08:42 DraFF.c
-rwxrwxrwx. 1 sanmayce sanmayce        18544 Dec 31 08:42 DraFF.elf
-rw-r--r--. 1 sanmayce sanmayce 104233480802 Oct 12 21:04 enwiki-20241001-pages-articles.xml
-rw-r--r--. 1 sanmayce sanmayce      2308563 Jan 29  2024 kazahana.c
-rwxr-xr-x. 1 sanmayce sanmayce       166840 Jan  5 19:29 kazahana_clang_18.1.6.x64.dynamic.elf
-rwxr-xr-x. 1 sanmayce sanmayce       154864 Jan  5 19:29 kazahana_gcc_14.1.1.x64.dynamic.elf
-rw-r--r--. 1 root     root          2565952 Jan  5 17:49 Kazahana_SS.txt
-rw-r--r--. 1 root     root              732 Jan  5 19:26 make_ELF.sh
drwxr-xr-x. 2 sanmayce sanmayce         4096 Jan  4 22:20 omp
-rwxrwxrwx. 1 sanmayce sanmayce         1520 Jan  5 19:40 runbench.sh
[sanmayce@djudjeto SS_vs_enwiki]$ su
Password:
[root@djudjeto SS_vs_enwiki]# sh runbench.sh
Architecture:             x86_64
  CPU op-mode(s):         32-bit, 64-bit
  Address sizes:          48 bits physical, 48 bits virtual
  Byte Order:             Little Endian
CPU(s):                   16
  On-line CPU(s) list:    0-15
Vendor ID:                AuthenticAMD
  BIOS Vendor ID:         Advanced Micro Devices, Inc.
  Model name:             AMD Ryzen 7 PRO 7840U w/ Radeon 780M Graphics
    BIOS Model name:      AMD Ryzen 7 PRO 7840U w/ Radeon 780M Graphics   None CPU @ 3.3GHz
    BIOS CPU family:      107
    CPU family:           25
    Model:                116
    Thread(s) per core:   2
    Core(s) per socket:   8
    Socket(s):            1
    Stepping:             1
    CPU(s) scaling MHz:   39%
    CPU max MHz:          5132.0000
    CPU min MHz:          400.0000
    BogoMIPS:             6587.74
    Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good amd_lbr_v2
       nopl xtopology nonstop_tsc cpui
                          d extd_apicid aperfmperf rapl pni pclmulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a
       misalignsse 3dnowprefetch osvw
                          ibs skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3 cdp_l3 hw_pstate ssbd mba perfmon_v2 ibrs ibpb stibp ibrs_enhanced vmmcall fsgsbase bmi1 avx2
       smep bmi2 erms invpcid cqm rd
                          t_a avx512f avx512dq rdseed adx smap avx512ifma clflushopt clwb avx512cd sha_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local
       user_shstk avx512_bf16 clzero i
                          rperf xsaveerptr rdpru wbnoinvd cppc arat npt lbrv svm_lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold v_vmsave_vmload vgif x2avic v_spec_ctrl
       vnmi avx512vbmi umip pku ospk
                          e avx512_vbmi2 gfni vaes vpclmulqdq avx512_vnni avx512_bitalg avx512_vpopcntdq rdpid overflow_recov succor smca fsrm flush_l1d amd_lbr_pmc_freeze
Virtualization features:
  Virtualization:         AMD-V
Caches (sum of all):
  L1d:                    256 KiB (8 instances)
  L1i:                    256 KiB (8 instances)
  L2:                     8 MiB (8 instances)
  L3:                     16 MiB (1 instance)
NUMA:
  NUMA node(s):           1
  NUMA node0 CPU(s):      0-15
Vulnerabilities:
  Gather data sampling:   Not affected
  Itlb multihit:          Not affected
  L1tf:                   Not affected
  Mds:                    Not affected
  Meltdown:               Not affected
  Mmio stale data:        Not affected
  Reg file data sampling: Not affected
  Retbleed:               Not affected
  Spec rstack overflow:   Mitigation; Safe RET
  Spec store bypass:      Mitigation; Speculative Store Bypass disabled via prctl
  Spectre v1:             Mitigation; usercopy/swapgs barriers and __user pointer sanitization
  Spectre v2:             Mitigation; Enhanced / Automatic IBRS; IBPB conditional; STIBP always-on; RSB filling; PBRSB-eIBRS Not affected; BHI Not affected
  Srbds:                  Not affected
  Tsx async abort:        Not affected
# dmidecode 3.6
Getting SMBIOS data from sysfs.
SMBIOS 3.3.0 present.

Handle 0x0016, DMI type 1, 27 bytes
System Information
       Manufacturer: LENOVO
       Product Name: 21K5001JUS
       Version: ThinkPad P14s Gen 4
       Serial Number: PF4PGB5F
       UUID: d4052ccc-350d-11b2-a85c-aceaa8d92359
       Wake-up Type: Power Switch
       SKU Number: LENOVO_MT_21K5_BU_Think_FM_ThinkPad P14s Gen 4
       Family: ThinkPad P14s Gen 4

# dmidecode 3.6
Getting SMBIOS data from sysfs.
```

```
SMBIOS 3.3.0 present.

Handle 0x0008, DMI type 17, 92 bytes
Memory Device
	Array Handle: 0x0005
	Error Information Handle: 0x0007
	Total Width: 32 bits
	Data Width: 32 bits
	Size: 16 GB
	Form Factor: Other
	Set: None
	Locator: DIMM 0
	Bank Locator: P0 CHANNEL A
	Type: LPDDR5
	Type Detail: Synchronous Unbuffered (Unregistered)
	Speed: 6400 MT/s
	Manufacturer: Micron Technology
	Serial Number: 00000000
	Asset Tag: Not Specified
	Part Number: MT62F4G32D8DV-026 WT
	Rank: 2
	Configured Memory Speed: 6400 MT/s
	Minimum Voltage: 0.5 V
	Maximum Voltage: 0.5 V
	Configured Voltage: 0.5 V
	Memory Technology: DRAM
	Memory Operating Mode Capability: Volatile memory
	Firmware Version: Unknown
	Module Manufacturer ID: Bank 1, Hex 0x2C
	Module Product ID: Unknown
	Memory Subsystem Controller Manufacturer ID: Unknown
	Memory Subsystem Controller Product ID: Unknown
	Non-Volatile Size: None
	Volatile Size: 16 GB
	Cache Size: None
	Logical Size: None

Handle 0x000B, DMI type 17, 92 bytes
Memory Device
	Array Handle: 0x0005
	Error Information Handle: 0x000A
	Total Width: 32 bits
	Data Width: 32 bits
	Size: 16 GB
	Form Factor: Other
	Set: None
	Locator: DIMM 0
	Bank Locator: P0 CHANNEL B
	Type: LPDDR5
	Type Detail: Synchronous Unbuffered (Unregistered)
	Speed: 6400 MT/s
	Manufacturer: Micron Technology
	Serial Number: 00000000
	Asset Tag: Not Specified
	Part Number: MT62F4G32D8DV-026 WT
	Rank: 2
	Configured Memory Speed: 6400 MT/s
	Minimum Voltage: 0.5 V
	Maximum Voltage: 0.5 V
	Configured Voltage: 0.5 V
	Memory Technology: DRAM
	Memory Operating Mode Capability: Volatile memory
	Firmware Version: Unknown
	Module Manufacturer ID: Bank 1, Hex 0x2C
	Module Product ID: Unknown
	Memory Subsystem Controller Manufacturer ID: Unknown
	Memory Subsystem Controller Product ID: Unknown
	Non-Volatile Size: None
	Volatile Size: 16 GB
	Cache Size: None
	Logical Size: None

Handle 0x000E, DMI type 17, 92 bytes
Memory Device
	Array Handle: 0x0005
	Error Information Handle: 0x000D
	Total Width: 32 bits
	Data Width: 32 bits
	Size: 16 GB
	Form Factor: Other
	Set: None
	Locator: DIMM 0
	Bank Locator: P0 CHANNEL C
	Type: LPDDR5
	Type Detail: Synchronous Unbuffered (Unregistered)
	Speed: 6400 MT/s
	Manufacturer: Micron Technology
	Serial Number: 00000000
	Asset Tag: Not Specified
```

```
	Part Number: MT62F4G32D8DV-026 WT
	Rank: 2
	Configured Memory Speed: 6400 MT/s
	Minimum Voltage: 0.5 V
	Maximum Voltage: 0.5 V
	Configured Voltage: 0.5 V
	Memory Technology: DRAM
	Memory Operating Mode Capability: Volatile memory
	Firmware Version: Unknown
	Module Manufacturer ID: Bank 1, Hex 0x2C
	Module Product ID: Unknown
	Memory Subsystem Controller Manufacturer ID: Unknown
	Memory Subsystem Controller Product ID: Unknown
	Non-Volatile Size: None
	Volatile Size: 16 GB
	Cache Size: None
	Logical Size: None

Handle 0x0011, DMI type 17, 92 bytes
Memory Device
	Array Handle: 0x0005
	Error Information Handle: 0x0010
	Total Width: 32 bits
	Data Width: 32 bits
	Size: 16 GB
	Form Factor: Other
	Set: None
	Locator: DIMM 0
	Bank Locator: P0 CHANNEL D
	Type: LPDDR5
	Type Detail: Synchronous Unbuffered (Unregistered)
	Speed: 6400 MT/s
	Manufacturer: Micron Technology
	Serial Number: 00000000
	Asset Tag: Not Specified
	Part Number: MT62F4G32D8DV-026 WT
	Rank: 2
	Configured Memory Speed: 6400 MT/s
	Minimum Voltage: 0.5 V
	Maximum Voltage: 0.5 V
	Configured Voltage: 0.5 V
	Memory Technology: DRAM
	Memory Operating Mode Capability: Volatile memory
	Firmware Version: Unknown
	Module Manufacturer ID: Bank 1, Hex 0x2C
	Module Product ID: Unknown
	Memory Subsystem Controller Manufacturer ID: Unknown
	Memory Subsystem Controller Product ID: Unknown
	Non-Volatile Size: None
	Volatile Size: 16 GB
	Cache Size: None
	Logical Size: None

Setting CPU to performance mode...
performance
 ____  __.                  .__
|    |/ _|____  ____________ |  |__ _____    ____ _____
|      < \__  \ \___   /\__  \ |  |  \\__  \  /    \\__  \
|    |  \ / __ \_/    /  / __ \|   Y  \/ __ \|   |  \/ __ \_
|____|__ (____  /_____ \(____  /___|  (____  /___|  (____  /
        \/    \/      \/     \/     \/     \/     \/     \/
Kazahana, a typhoon-class exact & wildcards & Levenshtein Distance (Wagner-Fischer) searcheress, r. 1-+
	+fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix_DEFINE_Trolldom_LineByLine_fix_BackToBuffer_Lowercasing-fix, copyleft Kaze 2024-Jan-29.
WILDCARD_IP_flag = 0
WILDCARD_FAST_flag = 2
Exact_flag = 0
EXHAUSTIVE_flag = 1
Enforcing Case-Insensitive Fuzzy (EXHAUSTIVE) mode ...
Pattern: Silvestor Staloune
omp_get_num_procs( ) = 16
omp_get_max_threads( ) = 16
Enforcing HEXADECAD i.e. hexadecuple-threads ...
Allocating Master-Buffer 458752KB ... OK
/; Speed: 00,000,921,828 bytes/second; Traversed: 103,817,284,332 bytes; Dumped: 3,568
Kazahana: Total/Checked/Dumped xgrams: 1,510,257,125/727,171,725,391/3,568
Kazahana: Performance: 897 KB/s
Kazahana: Performance: 13,321 xgrams/s
Kazahana: Performance: Total/fread() clocks: 113,367,028,289/129,976,490
Kazahana: Performance: I/O time, i.e. fread() time, is 0 percents
Kazahana: Done.

 Performance counter stats for './kazahana_clang_18.1.6.x64.dynamic.elf 4e Silvestor Staloune enwiki-20241001-pages-articles.xml 458752':

     114,386,981.25 msec task-clock                       #   14.968 CPUs utilized
          1,912,591      context-switches                 #   16.720 /sec
             20,118      cpu-migrations                   #    0.176 /sec
            115,629      page-faults                      #    1.011 /sec
430,253,392,918,667      cycles                           #    3.761 GHz                      (71.43%)
 32,760,411,949,501      stalled-cycles-frontend          #    7.61% frontend cycles idle     (71.43%)
```

```
643,563,996,279,834      instructions                     #    1.50  insn per cycle
                                                          #    0.05  stalled cycles per insn     (71.43%)
 99,305,407,571,701      branches                         #  868.153 M/sec                       (71.43%)
  1,069,396,561,488      branch-misses                    #    1.08% of all branches             (71.43%)
216,396,837,380,793      L1-dcache-loads                  #    1.892 G/sec                       (71.43%)
    804,760,064,684      L1-dcache-load-misses            #    0.37% of all L1-dcache accesses   (71.43%)
    <not supported>      LLC-loads
    <not supported>      LLC-load-misses

    7641.855604763 seconds time elapsed

  113257.808166000 seconds user
     110.154350000 seconds sys


 ____  __.                  .__
|    |/ _|____  ________ _____  |  |__ _____    ____ _____
|      < \__  \ \___   /\__  \ |  |  \\__  \  /    \\__  \
|    |  \ / __ \_/    /  / __ \|   Y  \/ __ \|   |  \/ __ \_
|____|__ \(____  /_____ \(____  /___|  (____  /___|  (____  /
        \/     \/      \/     \/     \/     \/     \/     \/
Kazahana, a typhoon-class exact & wildcards & Levenshtein Distance (Wagner-Fischer) searcheress, r. 1-+
       +fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix_DEFINE_Trolldom_LineByLine_fix_BackToBuffer_Lowercasing-fix, copyleft Kaze 2024-Jan-29.
WILDCARD_IP_flag = 0
WILDCARD_FAST_flag = 2
Exact_flag = 0
EXHAUSTIVE_flag = 1
Enforcing Case-Insensitive Fuzzy (EXHAUSTIVE) mode ...
Pattern: Silvestor Staloune
omp_get_num_procs( ) = 16
omp_get_max_threads( ) = 16
Enforcing HEXADECAD i.e. hexadecuple-threads ...
Allocating Master-Buffer 458752KB ... OK
/; Speed: 00,000,729,547 bytes/second; Traversed: 103,817,284,332 bytes; Dumped: 3,568
Kazahana: Total/Checked/Dumped xgrams: 1,510,257,125/727,171,725,391/3,568
Kazahana: Performance: 710 KB/s
Kazahana: Performance: 10,549 xgrams/s
Kazahana: Performance: Total/fread() clocks: 143,165,681,239/53,545,247
Kazahana: Performance: I/O time, i.e. fread() time, is 0 percents
Kazahana: Done.

 Performance counter stats for './kazahana_gcc_14.1.1.x64.dynamic.elf 4e Silvestor Staloune enwiki-20241001-pages-articles.xml 458752':

     144,354,040.79 msec task-clock                       #   14.511 CPUs utilized
          2,347,473      context-switches                 #   16.262 /sec
             21,606      cpu-migrations                   #    0.150 /sec
            115,448      page-faults                      #    0.800 /sec
587,891,209,932,717      cycles                           #    4.073 GHz                         (71.43%)
 20,393,559,085,655      stalled-cycles-frontend          #    3.47% frontend cycles idle        (71.43%)
660,378,518,227,361      instructions                     #    1.12  insn per cycle
                                                          #    0.03  stalled cycles per insn     (71.43%)
 90,380,842,631,801      branches                         #  626.105 M/sec                       (71.43%)
  1,184,652,928,058      branch-misses                    #    1.31% of all branches             (71.43%)
183,158,508,386,059      L1-dcache-loads                  #    1.269 G/sec                       (71.43%)
  1,390,220,103,774      L1-dcache-load-misses            #    0.76% of all L1-dcache accesses   (71.43%)
    <not supported>      LLC-loads
    <not supported>      LLC-load-misses

    9947.697494712 seconds time elapsed

  143118.425563000 seconds user
      47.499657000 seconds sys

Searching using 16 threads for pattern 'Silvestor Staloune' ...
Thread 12: Match found at file offset 27268737: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 32220262: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 35662155: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 35577564: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 35586149: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 35592686: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 90135653: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 149338919: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 220293911: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 224741936: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 224748488: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 224754664: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 224774024: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 224818253: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 224829334: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 224848568: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 224848726: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 372681172: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 372681191: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 411193281: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 411193978: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 419844711: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 437397178: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 436110245: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 436134544: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 443180915: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 8: Match found at file offset 443264367: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 479163547: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 509061499: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 509062056: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 509063327: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 509364266: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 619686494: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 680889633: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 774905873: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 807411696: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 836529468: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 831496719: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 831497023: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 896655122: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 899994633: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 943185511: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 943186740: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 943186781: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 943418161: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 947599906: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 947599955: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 947601463: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 943570452: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 987740562: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 993825125: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 1029745836: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 1049241096: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 1076804324: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 1097665460: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 1107955622: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 1109006185: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1120647343: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1120647462: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1148222349: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153499059: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153499174: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153500347: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153508203: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153508276: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153509099: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153514050: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153514601: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153544135: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153545559: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153545907: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153547128: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153547860: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153548186: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153548798: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153566728: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153577396: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153577857: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1153579448: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 1234787080: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 1281840427: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 1280252118: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 1316573046: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 1335073544: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 1335112352: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 1336264178: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 1346121913: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 1423279764: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451448193: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451449011: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451449053: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451449553: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451450074: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451450181: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451450378: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451450819: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451451310: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451451686: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451452493: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451454919: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451455421: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451456241: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451457806: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451458584: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451460230: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451461972: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451462702: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451463964: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451465002: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451467451: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451468407: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451472204: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451473651: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451474400: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451475023: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 5: Match found at file offset 1451482422: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451489161: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451490162: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451491815: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451493088: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451501286: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451503837: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451506429: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451507802: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451508467: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451511760: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451512473: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451513536: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451514118: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451514427: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451516818: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451519521: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451519973: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451520932: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451521283: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451523270: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451525334: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451528386: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451528901: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451530439: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451531677: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451532305: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451534562: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451535235: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451535898: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451536654: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451537648: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451538367: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451539759: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451541311: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451542108: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451544753: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451548826: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451549788: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451551746: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451552341: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451553651: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451554371: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451555729: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451558625: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451559660: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451561205: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451563303: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451564000: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451564766: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451565372: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451565930: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451566583: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451567275: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451568096: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451568544: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451569904: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451571631: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451572615: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451574062: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451575232: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451576548: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451576937: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451577012: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451578093: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451578384: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451578485: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451578515: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451579339: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1451882477: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 1448980514: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 1448996429: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 1489580470: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 1489592057: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 1482410224: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 1501844445: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 1501844588: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 1501846598: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 1501849205: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 1501867611: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 1501868476: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 1512573209: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1607151702: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 1610761267: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 1610817061: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 1610817675: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 1656318849: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 1667472107: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 1673721689: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 2: Match found at file offset 1664556066: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 1671018486: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 1694175698: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 1703631604: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 1700568481: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 1738707390: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 1739515078: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 1739518597: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 1748668190: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 1802539595: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 1800712215: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 1800713729: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 1800716953: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 1806300173: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 1818704796: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 1823049916: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 1823056144: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 1823113210: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 1834330472: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 1844162897: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 1844172000: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 1886572205: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 1924859606: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 1951596430: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 1951613304: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 1977251865: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 1989835717: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 2006096831: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2013204854: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029614658: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029614897: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029615014: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029615103: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029615158: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029616564: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029624740: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029628409: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029635021: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029637452: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029638274: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029639089: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029642622: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029643263: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029651174: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029657305: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029658158: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029658597: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029659639: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029660514: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 2029661226: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 2069532022: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 2069532208: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 2088831074: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 2088831343: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 2089214744: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 2089214967: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 2110494772: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 2170472150: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 2321977194: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 2332979432: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 2332979825: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 2360998355: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 2390403428: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 2413546213: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 2521497517: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 2538333926: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 2577018892: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 2588830300: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 2622372747: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 2622414341: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 2630255483: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 2630255648: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 2645871566: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 2648226168: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686245812: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686245981: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686246098: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686246171: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686247039: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686253550: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686257980: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686267746: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686268554: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686269455: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 2686269605: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 2709835805: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 2731258661: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 2731258832: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 2731258950: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 11: Match found at file offset 2731259023: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731260032: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731268292: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731271863: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731283702: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731288368: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731290694: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731294908: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731296104: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2731296254: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745901323: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745901414: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745901470: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745902620: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745909474: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745916912: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745917487: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745922288: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745922603: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745922726: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745924395: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2745925266: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 2751976428: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 2783039865: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 2803979646: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 2816580739: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 2818227454: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 2818458578: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 2834501496: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 2854236044: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 2858443268: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 2858443460: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 2861864733: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 2861867878: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 2861869046: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 2861872333: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2860920915: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 2858355319: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2962421907: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 2958467358: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 2955807652: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 2959992887: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 2980060049: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2967330051: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 2967330241: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 2977884493: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 2990304194: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 2988438480: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 3004218632: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 3001345172: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 3001345944: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003478018: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003478138: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003479253: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003483128: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003483751: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003485777: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003490871: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003491533: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003491959: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003492514: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003499055: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003501009: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003504539: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003507141: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003507584: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003507868: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3003508455: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 3008787758: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 3026492067: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 3039883247: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 3039969683: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 3042460721: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 3054552100: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 3055253741: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 3073471530: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 3104147494: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 3145520534: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 3164762634: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 3153855690: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 3177378459: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 3182582320: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 3182582530: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 3182582930: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 3233007086: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 3240784399: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 3259265602: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 3255485204: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 3255498383: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 7: Match found at file offset 3255538880: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 3255540336: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 3255547117: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 3255557937: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 3275293058: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 3326815532: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 3326815804: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 3354840640: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 3379659185: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 3379663059: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 3379670620: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 3379328932: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 3413044019: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 3443453681: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 3443506047: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 3443522371: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 3443522483: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 3477092094: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 3500514026: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 3486410564: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 3486418251: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 3486420540: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 3486420636: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 3488516341: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 3514902314: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 3532601537: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 3540131475: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 3538228301: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 3562459170: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 3562459212: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 3562473317: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 3584100928: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 3584125652: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 3584126617: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 3584129618: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 3584133462: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 3584138809: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 3584139902: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 3610284686: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 3603299273: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 3722346946: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 3722347583: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 3746607813: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 3782107940: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 3830451349: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 3873654975: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 3884022125: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 3878614265: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 3878616790: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 3878620327: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 3878626820: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 3929022804: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 3947039447: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 3959292042: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 3996221046: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 3995975492: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 4062687783: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 4081234016: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 4095203992: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 4089252771: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 4111847033: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 4112260009: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 4151508988: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 4184974173: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 4175215349: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 4260179380: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 4299037455: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 4355192783: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 4345049547: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 4375975334: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 4377322063: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 4377322304: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 4430358381: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 4478508855: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 4476873407: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 4476972331: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 4506510956: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 4536562467: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 4604452321: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 4604452727: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 4639290564: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 4643920912: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 4680913836: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 4691910324: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 4701184014: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 4701272056: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 4727363707: 'Silvester Stallone' (Edit Distance: 3)
Thread 1: Match found at file offset 4727363790: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 4727364284: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 0: Match found at file offset 4758006958: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 4819565118: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811423578: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811424253: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811424831: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811425076: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811425325: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811425471: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811425540: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811425786: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811425854: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811442170: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811442561: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811443402: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811444747: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811445488: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811446273: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811446625: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811449046: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811450867: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811451694: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811472186: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811476268: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811477555: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4811479418: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 4833257555: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 4855511309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 4879855509: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 4908706108: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 4908706343: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 4908947650: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 4908954680: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 4899618378: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 4899618960: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 4939550173: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 4975855846: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 4999208372: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 4996403406: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5025724289: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5025724320: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 5057084784: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 5103004170: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 5166456880: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 5185289615: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 5185294036: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 5211788358: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 5218803262: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 5217164128: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 5227617373: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 5223923211: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 5263332250: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 5301218725: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 5301218780: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 5301221362: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 5363087681: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 5424896213: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 5418991487: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 5462131162: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5506637328: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5506764086: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5506784928: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5524482137: ‘Silvester Stallone’ (Edit Distance: 3)
Thread 12: Match found at file offset 5557643260: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 5615564191: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 5615577080: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5646021620: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5646022966: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5646024654: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 5660779617: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 5673579417: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 5673579454: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 5673580018: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 5685750879: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 5710895042: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5702643266: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5702647113: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5713304290: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5713304320: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5713305809: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5713308357: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 5712532445: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 5736875519: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 5736875556: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 5736876801: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 5736880807: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5734868967: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5734868997: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5734870272: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 5740226439: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 5: Match found at file offset 5740227012: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 5740227049: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 5740228422: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 5740230631: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 5740231138: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 5740231531: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 5739256866: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 5736239145: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 5736241944: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5735247277: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 5747858956: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 5747859004: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 5747860636: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 5747863923: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 5734420489: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 5739618159: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 5739620611: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 5738658684: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 5738658721: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 5738659956: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 5738663063: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 5768904600: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 5810531403: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 5836903552: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 5849728826: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 5849737580: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 5853678557: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 5901178989: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943570327: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943571693: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943576566: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943576727: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943576773: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943577211: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943578366: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943598898: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943606098: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943610399: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943613078: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943613309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943613801: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943614041: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943614730: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943625162: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943626793: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943633995: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943635231: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 5943635518: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 5977650894: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 6104642808: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 6104643039: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131137066: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131137250: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131139317: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131145737: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131163436: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131163459: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131163488: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131164502: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131167240: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131170547: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 6131171830: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 6163786929: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 6163787116: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 6166498413: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 6191032566: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 6197914277: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 6221600933: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 6224126299: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 6224127662: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 6224140899: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 6224206345: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 6337814720: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 6385030495: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 6385032515: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 6385036960: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445714299: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445720862: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445969497: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445969558: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445970742: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445975758: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445978314: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445978351: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445982644: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445983517: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 6445984169: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 6488024854: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 6514316573: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 11: Match found at file offset 6514317062: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 6505181689: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 6591669587: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 6591669999: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 6617904116: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 6604923492: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 6604940672: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 6628657758: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 6628658984: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 6628678341: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 6655491054: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 6667576158: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 6667581828: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 6678588956: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 6678589163: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 6678591665: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 6678592159: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 6717662701: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 6716040262: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 6730683803: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 6747678065: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 6747689697: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6740863752: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 6745479762: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 6781446430: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 6794248826: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 6794255413: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 6794257157: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 6794257376: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 6789300605: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 6794272039: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6793271473: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 6794314747: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 6794315118: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 6790542806: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 6822494153: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 6932214288: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 6943069850: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 6964814164: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 6989680749: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 6989680947: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993582380: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993582550: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993583875: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993588710: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993591159: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993595528: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993612684: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993618419: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993619982: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993623175: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993623520: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993623847: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993624638: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993625129: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 6993625927: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 7025923725: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 7083523315: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7134232188: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 7162512220: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 7162512662: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 7162520507: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 7162520949: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 7177036553: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 7177039234: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 7199855179: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 7199855198: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 7199867144: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7195117946: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7211367563: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 7288786950: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 7293959872: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 7293976794: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 7293977377: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 7393531492: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7395623072: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 7400606347: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 7406979481: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 7406987080: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 7488976042: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 7488976104: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 7488976681: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494247339: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494257538: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494257688: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494259516: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494266062: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494267271: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494270936: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 12: Match found at file offset 7494273433: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494280493: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494280988: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494296900: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494301529: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494302419: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7494303663: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7536261745: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7536261840: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7536297316: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 7530443538: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 7530457088: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 7541928018: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 7545831836: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 7584293872: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 7630367622: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 7656508123: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 7686073161: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 7689219491: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 7681279160: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 7698796283: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7701301909: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7701303206: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7701308501: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7701315566: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7701316140: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 7701316961: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7762206257: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7762206856: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 7762207436: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 7811681213: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 7910497406: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 7913879341: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 7953885340: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 7953885908: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052060779: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052061598: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052065506: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052083833: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052083904: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052085203: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052086385: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052090532: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8052099340: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 8076177196: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 8114057732: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 8145848090: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 8145850075: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 8145855651: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 8145860228: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 8145861951: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 8166998040: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 8220190509: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268781732: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268783850: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268788908: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268796146: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268804959: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268819523: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268819767: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268820852: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268821425: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268822562: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268822935: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8268824155: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 8290958429: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 8316364961: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 8316365056: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 8316365079: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 8323336555: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 8340747891: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 8336791665: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 8353665195: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 8363024800: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 8359965154: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 8374316266: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 8374316692: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 8374319883: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 8374345467: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 8395113712: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 8395113874: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 8548186310: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 8567941287: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 8592534547: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 8605480757: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 8620200435: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 8678081149: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 8690867655: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 8688031105: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 1: Match found at file offset 8703991085: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 8720418839: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 8722595715: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8736878300: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8736878445: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8736880082: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8736884845: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8736885424: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8736903526: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8736906006: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8736907083: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 8744546530: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 8795781048: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 8795781516: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 8795782791: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 8795783656: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 8795787962: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 8795792266: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 8855239034: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 8854446214: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 8873501906: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 8884661311: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 8904610200: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 9047145446: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 9044226148: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 9037296952: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 9065875645: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 9072546421: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9074145495: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 9107444930: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 9109937910: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 9109939992: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 9138208714: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 9160728842: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 9160729042: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 9152707888: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 9214786533: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 9232067996: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 9224080171: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 9231406425: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 9310582761: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 9345902118: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9360107007: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9360107617: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9360108750: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9360109099: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9360109460: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9360111254: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9360116259: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9360117580: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 9369866994: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 9376713012: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 9395802238: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 9462444456: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 9459092578: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 9495231871: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 9489089806: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 9500620659: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 9488404533: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 9488417242: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 9509299131: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 9509302173: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 9536758077: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 9582632036: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 9582638257: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 9595972597: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 9595973486: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 9603508098: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 9603508474: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 9705361278: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 9747025535: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 9747027245: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 9814185316: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 9848357176: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 9848867991: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 9867672537: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 9878812160: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 9878860886: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 9935421549: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 9944564044: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 9959248473: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 9999596610: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 10021672088: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 10021672270: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 10057360907: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10055426589: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 10077059301: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 10087532733: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10095859493: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 15: Match found at file offset 10095859570: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10095860929: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10095863836: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10095873894: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10095874844: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10095875705: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10096121084: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 10188137830: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201672160: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201672373: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201672450: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201672506: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201673953: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201675348: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201676407: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201680649: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201681504: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201687371: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201701294: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201701822: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 10201702132: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10218850290: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10351113467: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10351117287: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 10356493190: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 10356493423: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10364873008: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10364887926: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10364887977: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10364888247: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10364888614: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10364890572: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10364890649: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 10370260952: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 10370263939: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 10381910201: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10399093348: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10409842213: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10427665469: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10427668837: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10447106592: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10459551515: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10467181501: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10478809695: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10478820882: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10493535072: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 10498410448: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 10522992515: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10544323450: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10544323475: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10544323722: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10544323958: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 10571745161: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10571429818: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10582185448: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10582186594: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 10587381833: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10582191318: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10582193630: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10582200310: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10582200545: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 10591559925: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 10605012964: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10614250731: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10614251529: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10614272619: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10614297141: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10614309415: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 10622827857: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10714984904: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 10756046835: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10802630015: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 10802630114: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 10802380413: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 10817376964: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 10838990378: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 10875210911: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10879287548: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 10879319412: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 10905871550: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 10905873559: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10931357789: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10931360293: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10931361327: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10931365130: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 10931370309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 10949059513: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 10960519792: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 10976394328: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 0: Match found at file offset 10996823022: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 11045689106: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 11057665098: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 11057665749: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 11060041392: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11139644503: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11139644533: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11170922378: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11170925433: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11170937876: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 11175286504: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 11175286644: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 11175287624: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 11175289879: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 11175294131: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 11189272533: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 11229019910: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 11229020462: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 11244396867: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 11239484120: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 11278168719: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 11287878323: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 11345866280: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 11345503695: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477674635: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477675530: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477676783: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477680281: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477682259: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477683580: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477685320: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477687051: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11477687785: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 11509258089: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11534028787: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11534030182: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11534036215: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11534038660: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11534039521: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11534042381: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11534059631: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11534061811: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 11562648616: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 11562649332: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 11552695906: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 11628447706: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 11639686791: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11667717842: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 11667718596: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 11714439366: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 11743293798: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 11737342401: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 11791327615: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 11790618230: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 11790618309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 11787976692: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 11798574181: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 11802194210: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 11802195851: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 11834061843: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 11926371166: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 11949337375: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 12046726230: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 12060910026: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 12075672269: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 12104308577: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 12111875130: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 12140353738: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 12193997320: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 12196118623: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 12197912121: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 12207332674: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 12259892125: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294659584: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294659756: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294660816: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294662977: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294663616: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294666385: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294669301: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294674101: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294674564: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294674891: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294681590: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294681636: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294683105: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294683360: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294684779: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 12294685409: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 10: Match found at file offset 12311365855: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12465742227: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 12459846625: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12477212900: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 12504888883: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 12504889023: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 12504891384: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 12504891473: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12514317247: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12581023107: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12581024016: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12581028626: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12581031221: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12581037339: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12581038300: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 12593933972: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 12595190589: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 12673315856: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 12673317282: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 12673322265: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 12673323157: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 12673325382: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 12673326244: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 12673327759: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 12673331040: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 12767852397: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 12775065612: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 12778656582: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12777679321: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12777681155: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12777682802: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12777687424: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12777690181: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12777690515: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12777703716: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12777709117: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 12781024787: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 12791207039: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12810932075: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 12813113016: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12827230412: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12827241644: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12827246078: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12827248283: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12827249451: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12827249509: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12827250061: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 12827250842: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 12833906709: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827218804: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827219417: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827220359: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827220932: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827221175: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827222033: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827223143: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827223535: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827223933: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827225518: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827226339: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 12827227393: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844447056: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844447537: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844515688: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844516334: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844517111: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844518278: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844518876: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844522778: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 12844522921: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847718267: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847718669: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847718725: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847720078: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847724645: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847729219: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847730895: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847731466: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847738023: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12847738805: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 12888731904: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 12904954985: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 12906882449: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12933101464: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12933102963: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12933103334: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 12933104826: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 12956978031: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 13004897537: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 14: Match found at file offset 13041653319: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 13166042064: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 13166042532: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 13208294501: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 13288571947: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 13300096379: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 13334961513: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 13333152157: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 13333153234: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 13333158895: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 13333167529: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 13333168873: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 13333169119: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 13333170689: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 13333171218: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 13355194941: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 13393185594: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 13490858528: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 13493684975: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 13539400916: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 13606894103: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598788774: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598790115: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598790373: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598795078: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598798849: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598803956: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598804812: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598806718: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598811454: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598812327: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598812454: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13598812883: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 13616065522: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 13650743990: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 13696471588: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13731204510: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 13804873331: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 13816836747: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 13961875691: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971250475: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971250851: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971250891: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971251085: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971251140: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971252442: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971259035: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971261673: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971279442: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971280368: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971289322: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971289568: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971290500: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971291019: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971292151: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971293258: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971293694: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971294177: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 13971294670: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 14003751484: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 14004274302: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 14002491368: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14065855062: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 14156950988: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 14229847520: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 14258585864: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 14255962077: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 14255988031: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 14281720548: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 14324623938: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 14324263047: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 14324374763: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 14324582882: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14436432828: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 14470897190: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 14482016495: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14503360288: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 14545517395: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 14535317346: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 14597905193: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 14597914247: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14602958496: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14602999420: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14603001889: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14603003012: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 14624804907: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 14647199574: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 14652556743: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 14776362698: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 15: Match found at file offset 14794604611: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 14807038798: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 14808115048: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 14808115475: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 14808116256: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 14808118052: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 14808118750: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 14808120326: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 14808121224: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 14808123973: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 14818429532: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14858932197: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 14878545147: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 14897198202: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 14897440282: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 14910629112: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 14910630544: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 14910633787: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 14976877232: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 14974063688: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15024503011: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15024503192: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15024503733: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15024505608: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15024507641: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15024513732: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15024514142: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 15041639806: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 15041656337: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 15041656461: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 15041769420: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15092027881: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 15109134586: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15159197290: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 15181560359: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 15181561975: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15176796611: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15176798030: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 15171821788: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 15194473935: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 15196859384: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 15238691683: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 15247149264: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 15260152370: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15327411011: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 15424702620: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 15471583959: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 15471586886: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15482733044: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 15529533075: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 15535475444: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 15535478208: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15608743662: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 15633276568: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15628544536: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15628544579: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15628554099: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15666353260: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15666354124: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 15666359110: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 15674112238: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 15730379607: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 15780077901: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 15780077945: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15849055437: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15849109712: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15849131397: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 15873518909: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 15865630635: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 15881599615: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 15894408757: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 15981656698: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 16044294622: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 16118769129: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 16167636567: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 16192034024: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 16226514960: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 16235991595: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 16246850269: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 16253420678: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 16280326698: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 16302089093: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 16321319362: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 16321319805: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 16361098357: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 16392486094: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 16421915721: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 16412868792: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 16412868886: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 11: Match found at file offset 16456757238: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 16456757486: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 16473173742: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 16479364669: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 16509010740: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 16559614892: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 16604544339: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 16608894735: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 16681182753: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 16688025785: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 16706229548: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 16750490742: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 16758929023: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 16786297854: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 16837088647: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 16828442580: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 16880188773: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 16939986364: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 16932248861: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 16932256057: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 16932257054: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 16932259247: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 16932260147: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 16932261241: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 16932263200: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 16971435251: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 16971444456: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 16971444951: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 16985029204: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 16985031495: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 16985043517: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 16988749871: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 16996262847: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 17038049712: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 17076923491: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 17145554393: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 17295318779: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 17303048875: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 17303049277: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 17340209420: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 17340209880: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361273482: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361279215: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361286844: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361287162: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361291651: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361292784: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361296582: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361297023: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361300664: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 17361301371: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 17352475613: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 17359936937: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 17362851100: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 17361245575: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 17361246045: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 17361246574: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 17361247091: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 17361248750: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 17432109955: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 17470830145: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 17486310017: 'Silvester Stallone' (Edit Distance: 3)
Thread 13: Match found at file offset 17495693081: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 17512276902: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 17512278474: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 17519741396: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 17535081168: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 17603498681: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 17653731792: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 17704504353: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 17704508321: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 17732994521: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 17798129426: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 17804106009: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 17847474063: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 17884570609: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 17917610299: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 17917610419: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 17954850701: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 17954852812: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 17954857844: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 17954860906: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 17951337741: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 17954312769: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 17978456870: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18006859524: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 18045276804: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 18045280131: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18074656337: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 4: Match found at file offset 18074659393: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18074659504: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18074659648: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18074659747: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18074659979: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18074660816: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18176771509: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18176777792: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 18188110026: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 18261315732: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 18287192372: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 18320489721: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 18332842479: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 18332842660: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 18367775254: '>Sylvester Stalone' (Edit Distance: 4)
Thread 6: Match found at file offset 18367775255: 'Sylvester Stalone<' (Edit Distance: 4)
Thread 6: Match found at file offset 18367775338: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 18367775784: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 18377975648: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 18459722752: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 18459722934: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 18459723275: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 18487270290: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 18523428033: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 18539368739: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 18555374775: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 18620265677: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 18673373896: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 18688527434: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 18694963344: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18716053717: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18729200788: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18729201320: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18729201437: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18729201930: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18729203372: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18729205852: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 18729206639: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 18855660179: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 18871296508: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 18882855180: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 18931363348: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 18941525757: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 18967599850: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 18987884462: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 18987885302: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 18987889402: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18999894901: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18999898951: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18999899137: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 18999899266: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 19018996379: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 19219172383: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 19255099725: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 19247395426: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19263589150: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19263590279: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19263595242: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 19281859985: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 19294500935: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 19296897769: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 19332957211: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 19331263318: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 19369320392: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 19366107623: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 19373323783: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 19419713937: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 19419719854: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 19419722584: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 19419722829: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 19418122635: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 19419473478: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445020837: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445021067: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445021176: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445021230: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445022330: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445024872: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445025804: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445035952: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445036173: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 19445036349: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 19494595810: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 19494595829: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 19491889399: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 19521166863: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 19521184761: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 19521192012: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 19557111384: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 14: Match found at file offset 19542053818: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 19614264583: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 19735426875: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 19744650730: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 19863740107: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 19887497869: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 19959831661: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 19961613995: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 20088454708: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 20222767179: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 20309149304: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 20309153415: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 20344420921: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 20344438082: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 20352093134: ' Sylvester Stalone' (Edit Distance: 4)
Thread 9: Match found at file offset 20352093135: 'Sylvester Stalone.' (Edit Distance: 4)
Thread 7: Match found at file offset 20394865033: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 20394866743: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 20394870574: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 20410433492: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 20433467947: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 20564190912: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 20575620234: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 20585340334: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 20581384888: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 20608693145: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 20727810152: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 20769589286: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 20772712213: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 20772712235: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 20779389584: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 20799488401: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 20831018728: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 20823862634: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 20838974361: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 20861219553: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 20880856037: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 20912077487: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 20919650892: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 20969702290: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 20998399027: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 21007824342: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 21007832083: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 21157638842: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 21219245440: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 21254400070: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 21254400727: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 21254407820: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 21283121875: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 21283122285: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 21283122449: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 21295483861: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 21318801481: 'Silvester Stallone' (Edit Distance: 3)
Thread 2: Match found at file offset 21583770902: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 21598961211: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 21611647742: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 21653112823: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 21655508765: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 21656046348: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 21687920297: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 21728538360: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 21752921447: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 21798716609: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 21798716998: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 21798720809: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 21799791776: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 21838190442: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 21937920862: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 22051058689: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 22168563090: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 22195479900: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 22217997048: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 22254225103: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 22247317019: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 22247321170: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 22295977052: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 22398307014: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 22546936243: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 22659848761: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 22708829280: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 22712440680: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 22712441415: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 22756005526: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 22798963216: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 22885730352: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 22885730804: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 22885731434: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 22897696260: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 22923164325: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 3: Match found at file offset 22959710756: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 22949051763: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 22949052585: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 23003816247: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 23018356570: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 23018357248: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 23037209174: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 23052253902: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 23093512613: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 23086416394: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 23126318505: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 23147547191: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 23180926905: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 23185678745: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 23181911072: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 23244886286: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 23259780467: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23273134224: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 23361825622: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 23362187715: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 23375657987: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23402730377: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23402740683: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 23414085974: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 23433056138: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 23530398020: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 23556823085: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23622929415: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23622932614: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23622933197: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23672061550: ‘Salvestor Stallone’ (Edit Distance: 3)
Thread 15: Match found at file offset 23672061634: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23672061903: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 23672062127: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 23734369748: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 23734371723: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 23879355412: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 23940269682: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 23994737395: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 24001787351: ‘[Sylvester Stalone’ (Edit Distance: 4)
Thread 4: Match found at file offset 24001787352: ‘Sylvester Stalone]’ (Edit Distance: 4)
Thread 4: Match found at file offset 24001788497: ‘ Sylvester Stalone’ (Edit Distance: 4)
Thread 4: Match found at file offset 24001788498: ‘Sylvester Stalone ’ (Edit Distance: 4)
Thread 4: Match found at file offset 24001788791: ‘ Sylvester Stalone’ (Edit Distance: 4)
Thread 4: Match found at file offset 24001788792: ‘Sylvester Stalone ’ (Edit Distance: 4)
Thread 14: Match found at file offset 24045927928: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 24092747411: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 24144048316: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 24200294253: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 24241393721: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 24343497805: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 24342489485: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 24389936741: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 24389937268: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 24424667335: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 24424667607: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 24424667831: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524657220: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524658196: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524661602: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524661828: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524662339: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524662864: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524663046: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524663204: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524663647: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524663998: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524668051: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524673389: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524674387: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524677708: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524681848: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524682578: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524683492: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524683955: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524685387: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524685663: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524686653: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524687172: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524688297: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524688906: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524692452: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524693152: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524693739: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524696217: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524697291: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524699582: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524718751: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524721020: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 2: Match found at file offset 24524721513: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524727721: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524728370: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524728703: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524732523: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524733192: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524734615: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524734982: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524735527: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24524737134: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 24578698064: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 24578698184: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 24596647609: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 24627937079: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 24643985270: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 24645465850: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 24700348033: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 24719450412: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 24734298718: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 24734299360: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 24731494341: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 24731494618: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 24831596816: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 24824388637: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 24875534178: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 24891071385: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 24891078050: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 24937068371: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 24943026917: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 25040405597: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 25040405692: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25117257669: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 25140166257: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 25176093705: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 25203601081: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 25193276794: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 25231848638: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 25231851660: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 25246336219: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 25317975108: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 25317976757: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 25317976781: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 25388244830: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 25389885682: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 25412093220: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 25495214036: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 25574972002: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25563952361: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 25616207223: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 25647892883: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671422432: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671423194: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671423309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671424624: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671426947: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671432476: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671434311: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671436036: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671438163: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671438597: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671442130: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671442863: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671443038: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671444578: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671444746: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671444770: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25671444875: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 25691445255: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 25691453900: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 25691457668: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 25691459253: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 25691459567: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 25753354667: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 25771640520: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 25810014609: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 25903072637: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 25945827425: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 26047636180: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 26047636228: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 26047641201: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 26135951087: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 26196387009: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 26232003107: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 26335051651: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 26335051966: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 26335052191: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 26374832631: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 26443095952: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 26443096010: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 3: Match found at file offset 26442161302: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26442168828: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26442171544: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26442198391: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26442203105: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 26441266862: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 26444384915: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 26444384961: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 26444415364: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 26444415401: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 26441318486: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 26446516790: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 26446552645: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26442536552: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26442536598: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 26444635803: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 26444654478: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 26440553973: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 26440554029: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 26441617579: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 26441617616: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 26440606393: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 26440606439: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 26443764302: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 26443779266: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 26443991970: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 26443992023: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 26446070435: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 26446070472: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 26564199871: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26591911815: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26591911852: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26591958499: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593037081: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593037127: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594134994: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26592090119: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594190298: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594190335: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593211200: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593211278: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594286812: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594287090: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26592218975: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594353267: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26592292722: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26592292759: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594385155: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593609193: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26592585667: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594663918: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594663955: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 26592599036: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594708504: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594708541: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593681486: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 26591624265: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 26591624302: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594771464: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26594771510: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 26591671438: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 26591759087: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593855240: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593932340: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 26593943969: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 26805832759: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 26805903967: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 26879463524: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 26934875844: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 26958425637: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 27113274843: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 27216974957: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 27222154325: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 27221305858: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 27298884057: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 27298914569: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 27298925944: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 27411158821: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 27513872920: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 27514054296: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 27514092450: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 27514153297: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 27514403659: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 27514403683: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 27536872085: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 27541026629: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 27635525098: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 27651193247: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 27665624480: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 11: Match found at file offset 27665625531: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 27749497188: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788554356: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788560060: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788579277: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788585381: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788610324: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788660548: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788680416: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788682404: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788747041: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 27787131471: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788848586: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 27788848610: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 27853219382: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 27874376155: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 27888001899: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 27911338261: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 27953831817: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 28026514309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 28026514818: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 28026515026: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 28133461450: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153273277: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153273912: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153275159: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153275350: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153275372: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153277555: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153277830: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153279897: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153340206: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153340810: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153341787: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153344870: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153344891: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153345924: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153349112: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153349700: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153349944: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153350529: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153351912: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153355155: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153355864: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28153356032: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 28166843569: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 28166843795: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 28166844401: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 28229430761: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 28407302709: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28423290656: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 28441678398: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28457041173: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 28490906234: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28607499564: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 28607516046: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 28634362494: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 28634377690: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 28689134922: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 28732276452: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 28732280470: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 28878578840: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 28950164209: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 28950164442: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 28944750614: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 28948918739: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 29101381002: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 29249239823: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 29292850350: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 29310555474: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 29431039356: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 29453690566: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 29453695384: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 29509057089: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 29528879500: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 29528880929: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 29582915252: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 29607242513: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 29606479947: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 29642935855: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 29793369259: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 30054532748: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 30087968593: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 30113367921: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 30127339597: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 30186896892: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 30225086485: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 30269919673: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 30269937304: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 8: Match found at file offset 30269937792: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269938686: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269941738: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269942903: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269944306: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269945568: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269946822: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269948337: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269948616: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269949972: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269950926: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269951255: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269952037: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269952298: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269958367: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30269958439: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 30451990950: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 30498319718: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 30498378085: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 30522222604: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 30522222623: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30538090394: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 30549541738: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 30618785175: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 30618785711: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 30618787461: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 30618917173: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 30679968576: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 30720999421: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 30746751424: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 30771489816: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 30871243803: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 30917016510: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 31015926435: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 31126214161: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 31146246535: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 31146246553: '¦Sylvester Stalone' (Edit Distance: 4)
Thread 6: Match found at file offset 31146246554: 'Sylvester Stalone]' (Edit Distance: 4)
Thread 14: Match found at file offset 31166365304: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 31322514838: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 31409072640: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 31471186633: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 31471187177: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 31466117161: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 31466125194: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 31504442679: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 31549589563: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 31558117301: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 31576877527: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 31592588392: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 31701917045: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 31766010297: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 31820669790: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 31911347189: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 31947758054: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 31947758196: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 31954265405: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 31977099301: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 32056052290: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 32143096165: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 32218628963: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 32215850681: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 32375341059: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 32383139523: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 32383145320: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 32448547111: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 32448547924: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 32448552014: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 32448558963: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 32576279578: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 32589693976: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 32656735620: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 32683380056: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 32773637838: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 32798850584: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 32847521015: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 32894582121: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 32925035157: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 33013966375: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 33115964181: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 33115964469: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 33115964693: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 33228879781: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 33390149788: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 33390149934: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 33433522108: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 33433622867: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 33483224518: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 33600825069: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 12: Match found at file offset 33600838433: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600842963: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600843515: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600845597: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600845869: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600846967: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600847776: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600848820: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600849087: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600849507: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600850106: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600851678: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600853073: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600853351: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600853751: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600854480: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600855013: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600855442: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600855821: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600855841: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 33600856426: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 33676323313: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 33676323571: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 33676323691: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 33698583285: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 33698583396: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 33698079215: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 33770216388: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 33814923001: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 33814923217: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 33814928058: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 33838035029: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 33907106236: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 33907108239: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 33966154363: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111505377: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111505592: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111505831: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111507237: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111512663: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111541803: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111542901: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111543660: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111544031: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111544977: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111548737: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111549920: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111551842: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111553672: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111558914: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34111560128: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 34107918551: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 34193202562: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 34233545398: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 34249144192: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 34298454367: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 34424918421: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 34467293896: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 34529289871: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 34579122151: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34603967212: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 34628419889: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 34627582973: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 34690423574: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726158969: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726159193: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726159268: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726159480: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726163403: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726168870: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726174024: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726184346: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726185264: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726189470: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726192939: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726193161: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726193603: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726194034: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726194969: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726195631: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726196200: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34726196587: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 34761004951: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 34809070804: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34843872351: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34843874539: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 34843887081: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 34849212609: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 34863486723: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 12: Match found at file offset 34863486896: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 34931959078: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 35037103730: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 35164389298: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 35164389720: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 35164390900: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35191455633: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35191468713: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35191471046: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35191471984: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35191486566: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35191487755: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35191487940: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 35192570901: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 35234275471: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 35234276207: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 35234276857: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 35234230138: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 35234230807: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 35234237134: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 35364448746: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 35364450689: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 35397222180: ':Silvestr Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 35397222181: 'Silvestr Stallone]' (Edit Distance: 4)
Thread 7: Match found at file offset 35397222764: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 35397223350: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 35397223944: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 35397225431: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 35413673091: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 35413673195: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35426818830: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35426819442: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35426820037: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 35426820637: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 35576557041: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 35648122913: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 35652916772: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35808933978: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35808943013: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809062773: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809062797: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809154927: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809162684: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809163893: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809194219: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809198496: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809205937: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809218652: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809413620: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809419751: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809454905: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809463183: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809499479: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809520814: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 35809567801: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 35893156468: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 35951582078: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 35951582580: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 35951582619: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 35951582724: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 35951582816: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 36034940941: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 36045534300: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 36152097241: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 36163192896: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 36163198839: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 36251309553: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 36266714357: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 36268894997: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 36357676265: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 36514611118: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 36543512334: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 36564220020: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 36585212065: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 36587824028: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 36632646501: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 36792604573: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 36792605203: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 36877940852: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 36888390936: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 36919960642: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 36940086044: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 36962157545: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 36962197000: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034150426: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034150970: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034151389: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034151980: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034153195: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 0: Match found at file offset 37034154570: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034156337: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034162921: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034176723: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37034181021: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 37066386525: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073847632: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073847856: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073847946: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073848103: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073848255: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073849570: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073856647: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073858036: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073861167: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073864778: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073865417: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073866988: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073868082: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073871379: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073887675: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073888818: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073889658: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073890423: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 37073890535: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 37143983699: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 37198830533: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37268404400: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37268406305: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 37268433242: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 37289073617: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 37410410509: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 37458573314: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 37458578175: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 37459144163: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 37464120516: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 37506930250: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 37528270787: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 37565496451: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 37582369935: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 37582385357: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 37582395570: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 37629589754: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 37633305521: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 37733219577: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 37733219617: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 37731506726: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 37732562784: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 37732571739: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 37731683208: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 37733824000: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 37734943921: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 37732886745: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 37759111128: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 37783331164: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 37864805972: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 37864810547: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 37976085744: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 37983151565: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 38064495210: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 38119635037: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 38148045032: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 38148052946: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 38176745807: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 38176771978: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 38194707670: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 38194712186: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 38194725582: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 38194729034: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 38188741359: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 38211079734: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 38226747443: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 38406338092: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 38497778916: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 38489962155: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 38501788629: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 38501801796: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 38556340422: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 38556341114: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 38690890792: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 38690891169: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 38703661048: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 38703685754: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 38761357982: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 38816995270: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 38823374851: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 39046557181: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 39202727816: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 39234752173: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 9: Match found at file offset 39338923857: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 39362517107: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 39358812076: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 39414745658: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 39686517350: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 39697089238: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 39742939839: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 40002551340: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 40028816025: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 40080100528: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 40282763788: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 40306233427: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 40351753721: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 40380889979: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 40481318077: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 40574316408: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 40630627938: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 40779105255: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 40836963917: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 40836970500: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 40840486705: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 40840951900: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 40934808459: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 40934809308: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 40953102742: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 40953103425: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 41050149928: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 41050161300: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 41050161590: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 41184181898: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 41270013039: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 41301865295: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 41310179140: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 41317631502: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 41348161365: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 41390056736: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 41398376145: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 41394585975: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 41394586435: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 41398730598: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 41409231786: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 41499895707: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 41499895805: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 41499896309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 41504353530: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 41726151791: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 41728853113: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 41728853705: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 41744298012: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 41793612508: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 41853261170: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 41880120496: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 41919336658: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 41946902394: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 41947079205: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 41984579658: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42007257226: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42007258798: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42066223581: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42066223742: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42066234395: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 42231890102: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 42231892400: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 42262321650: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 42342625978: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 42378089229: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 42422246492: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 42464975329: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 42474383623: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 42544019400: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 42614549103: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 42614550280: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 42614550848: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 42614551434: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 42617991157: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 42729510637: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 42729510716: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 42984237759: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996635979: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996636003: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996688167: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996703216: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996704421: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996728152: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996785884: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996798230: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 42996802515: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 43013275879: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43081425954: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 15: Match found at file offset 43302560275: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 43302650066: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 43302650363: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 43320205617: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 43330391896: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 43416516431: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 43467939910: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 43468102908: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 43468240933: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 43524606019: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43709026606: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43727482664: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 43740582661: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43759394759: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43759400928: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43759402838: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43759425171: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43759445774: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 43759450590: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 43798032248: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 43797929844: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 43874812719: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 43874814728: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 43913214285: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 44170415407: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 44170498530: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 44329671184: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 44561839595: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 44704153734: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 44741167903: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 44741172877: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 44807304799: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 44861577355: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 44861580412: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 44962399916: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 45057369310: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 45220956316: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 45242462309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 45319190673: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 45368868887: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 45379855599: ‘Sylvestor Stallone’ (Edit Distance: 3)
Thread 13: Match found at file offset 45400052312: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 45400052928: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 45400053047: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 45417681238: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 45604979983: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 45604980816: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 45614384855: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 45614389324: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 45672334714: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 45672334886: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 45672335661: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 45722604845: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 45748073138: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 45777181217: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 45777294393: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 45777333388: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 45786708489: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 45855321254: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 45854410308: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 45854495618: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 45855517958: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 45853685817: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 45853849609: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 45851912613: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 45856012634: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 45859179876: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 45994213463: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 46049401068: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 46191028787: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 46338802277: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 46414639349: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 46465603663: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 46465888774: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 46616611284: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 46636998240: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 46739603107: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 46739611872: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 46900881093: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 47008910199: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 47192524623: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 47207250696: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 47248476054: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 47448966119: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 47472602215: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 47569401193: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 47596565161: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 47596567907: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 47590523683: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 9: Match found at file offset 47596843938: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 47691379286: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 47718184538: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 47794441009: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 47818460185: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 47819581295: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 47918918929: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 47918919863: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 47918919924: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 47918920128: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 47918920195: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 47918920294: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 47918920491: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 48046586589: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 48046586748: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 48067195514: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 48081095907: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 48185884316: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 48253980916: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 48274822159: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 48274207482: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 48436055143: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 48452169491: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48530333735: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 48524439238: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 48613069901: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 48731089264: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 48748050621: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 48834087041: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880736112: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880737461: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880740950: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880742254: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880742499: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880743403: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880743722: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880746238: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880751074: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 48880751285: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 48895057138: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 48904535894: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 48915774799: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 48915775289: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 48923053932: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 48922305072: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 49043300879: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 49057009999: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 49144131377: ‘=Silvester Stalone’ (Edit Distance: 3)
Thread 11: Match found at file offset 49144131378: ‘Silvester Stalone ’ (Edit Distance: 3)
Thread 0: Match found at file offset 49177203373: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 49291480406: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 49291480437: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 49291480542: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 49302567272: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 49302567440: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 49484328393: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 49518564566: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 49738987968: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 49812414348: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 49977957055: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 50090936548: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 50142576449: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 50142577099: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 50142577222: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 50142577363: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 50142578035: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 50182708083: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 50288644465: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 50330890363: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 50374604682: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 50502571270: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 50696670459: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 50743826339: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 50782758422: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 50782759136: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 50850939438: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 50839747636: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 50899182743: ‘ Sylvester Stalone’ (Edit Distance: 4)
Thread 6: Match found at file offset 50899182744: ‘Sylvester Stalone ’ (Edit Distance: 4)
Thread 6: Match found at file offset 50899183565: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 50943453425: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 51030858082: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 51150493890: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 51192736418: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 51192744866: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 51192745309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 51192746462: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 51324763497: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 51373843700: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 8: Match found at file offset 51373843804: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 51432141337: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 51463512257: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 51522122115: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 51567656862: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 51664974726: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692374025: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692374243: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692375295: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692375703: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692380847: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692381042: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692407435: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692431588: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692450099: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692469198: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692471965: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692474230: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 51692475087: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 51768069297: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 51778260956: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 51931778672: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 51931911405: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 51931911931: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 51932654673: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 52031914611: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 52072100113: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 52072100206: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 52114267223: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 52114268928: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 52114269801: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 52114270766: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 52114271725: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 52431947395: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 52431949702: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 52431999144: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 52431926288: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 52527568422: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 52597774795: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 52617145943: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 52615192438: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 52731988156: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 52731988211: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 52810803081: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 52903164828: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 52950716335: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 52950858053: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 53013350742: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 53034479074: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 53132413218: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 53638893254: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 53638893501: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 53675886103: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 53760263611: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 53760305179: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 53760442207: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 53810376278: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 53820551141: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 53820551165: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 53834407919: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 53834407942: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 53878052069: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 53878130481: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 53878206488: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 53872208038: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 54081235548: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108858914: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108860707: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108865752: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108873734: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108875063: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108879527: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108879852: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108880076: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 54108880156: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 54130834982: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 54121628366: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 54372719401: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 54494506732: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 54494507048: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 54494379611: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 54494397244: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 54494483037: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 54494483309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 54494493756: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 54693294475: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 54713032350: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 54713046494: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 54881449522: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 0: Match found at file offset 54899347023: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 54980298626: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 54998474149: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 55052908079: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 55204143260: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 55204143498: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 55308362289: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 55372569208: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 55404899222: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 55467208804: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 55487828291: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 55562094380: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 55568409561: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 55823279361: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 55873401401: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 55965373676: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 56221458542: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 56325874592: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 56325875264: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 56325875305: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 56430006165: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 56449502828: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 56509694673: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 56509694722: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 56509695506: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 56509696201: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 56509697069: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 56509699921: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 56794445396: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 56794445509: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 56794445545: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 56794485986: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 56794490758: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 56795067080: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 56795067555: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 56829365282: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 56882914309: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57018335693: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57018338312: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57032057462: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57032059748: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 57030883023: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 57032046711: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57068172668: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57068173350: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57068175253: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173325072: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173325201: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173325318: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173327886: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173328220: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173334223: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173338274: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173339571: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173343851: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173345913: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173347001: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173355017: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173355251: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173360062: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173371169: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173377407: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173384515: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 57173385432: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199381824: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199383482: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199384595: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199384643: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199384900: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199385219: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199385269: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199389259: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199392445: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199410538: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199411325: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199411896: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199412439: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199421104: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199421637: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57199424723: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 57309698282: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 57309834741: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 57309851345: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 57309862183: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57422352366: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57422352481: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57422353935: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57422358599: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 57422360810: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 0: Match found at file offset 57422361525: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57432274028: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 57435694735: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 57533900714: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 57533901706: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 57533902251: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 57533906806: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 57533908262: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 57532909936: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 57532910268: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 57532910515: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 57532910598: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57531138887: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57531140011: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 57582019710: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 57582028198: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 57645685441: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 57668589892: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 57739277709: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 57822661506: ‘[Sylvester Stalone’ (Edit Distance: 4)
Thread 3: Match found at file offset 57822661507: ‘Sylvester Stalone¦’ (Edit Distance: 4)
Thread 14: Match found at file offset 58029495023: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 58051494814: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152246370: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152247263: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152247571: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152247885: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152248992: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152252726: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152253554: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152254291: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152254721: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58152255160: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 58409159222: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 58478960725: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 58478965196: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 58478972301: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58788928731: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 58788929251: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 58803458798: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 58803459607: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 58803459699: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 58803460402: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 58803478612: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 58803478651: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 58803478720: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 58911099585: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 59009189643: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 59059963551: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 59108601616: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 59173327016: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 59261720039: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 59401880797: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 59394389579: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 59408609517: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 59468348194: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 59508180692: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 59534778002: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 59669518351: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 59661546233: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 59706020973: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 59898744266: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 59958250881: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 59958254714: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 59983678714: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 60044248178: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 60044248254: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 60125663983: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 60140532021: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 60187237324: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 60194867476: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 60204983295: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 60267225962: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 60362459048: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 60365463217: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474627808: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474627870: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474628990: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474629317: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474633180: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474634046: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474634602: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474635161: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474637044: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474639080: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474639415: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 60474639899: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 60540747073: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 60572113034: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 2: Match found at file offset 60571877739: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 60571944424: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 60766017144: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 60766017977: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 60766019098: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 60766019473: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 60770191314: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 60770192725: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 60963842714: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 60963848424: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 61020218454: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 61025634408: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 61025634427: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 61154128826: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 61149481423: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 61156880427: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 61276763429: ‘[Sylvester Stalone’ (Edit Distance: 4)
Thread 1: Match found at file offset 61276763430: ‘Sylvester Stalone]’ (Edit Distance: 4)
Thread 5: Match found at file offset 61317705947: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 61383990913: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 61458003434: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 61618333383: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 61646983574: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 61647076780: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 61637207671: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 61802085048: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 61802085095: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 61802085214: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 61802085260: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 61831290891: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 61882708595: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 61900864639: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 62034711512: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 62079195282: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 62115636452: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 62115636778: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 62635303052: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 62658275316: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 62683694980: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 62701804638: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 62784900319: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 62845310739: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 63044514385: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 63067949535: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 63080147762: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 63087505483: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 63338410189: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 63335573362: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63453177882: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63453179842: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63453198942: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63453236039: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63453358298: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63453361285: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 63568345089: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 63603634841: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 63655843594: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 63890082174: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 63916913055: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991667057: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991667158: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991667382: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991668927: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991670373: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991676031: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991676912: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991681777: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991682760: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991684273: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991685368: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991687070: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991687643: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991688192: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991689946: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991690128: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991690906: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991692100: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991692823: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991693196: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991693476: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991693887: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991694195: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991694978: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991699649: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991701592: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991702104: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991702630: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991705216: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991707291: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 1: Match found at file offset 63991715190: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991718544: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991733928: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991736763: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991736985: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991737134: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991738736: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991739509: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991739897: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991740556: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991741020: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991741598: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 63991742727: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 64006338690: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64078248002: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 64156203476: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 64361631641: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461788042: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461788690: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461788806: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461789663: ‘Silvestro Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461790802: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461792330: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461793061: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461793745: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64461795851: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 64699209500: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 64780095477: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 64786171969: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 64786172395: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 64786172955: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 64786173486: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 64806577609: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 64806577682: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 64806578386: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 64883088990: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 64889175463: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 64965006299: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968501714: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968502205: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968502384: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968502464: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968504697: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968506013: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968508961: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968514422: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968517999: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968519205: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968519445: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968519870: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968520449: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968521298: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968522673: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968522964: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968526600: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968530292: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968541492: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968542572: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968544871: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968545166: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968545396: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968545676: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968546179: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968547427: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968547963: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968548058: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968548660: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968548752: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968549991: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968552711: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968553502: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968553807: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968553890: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968554246: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968554707: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968555120: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968555630: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968556919: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968557550: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968557660: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968557820: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968561792: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968562963: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968563570: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 64968565855: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 65071101162: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 65095888729: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 65095891025: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 65095900041: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 7: Match found at file offset 65095909366: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 65095909593: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 65256782441: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 65370873076: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 65484100635: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 65484100889: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 65484101101: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 65561416147: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 65561420949: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 65566855936: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 65619527841: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 65668514201: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 65691111247: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 65707787782: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 65707910918: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 65707911133: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 65707911628: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 65759379343: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 65759407218: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 65940314957: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 66090009221: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 66237526722: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 66303316532: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 66303316681: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 66303325582: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 66641554389: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 66725412006: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 66725537229: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 66783735115: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 67148410351: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 67208523022: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 67204243010: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 67204246588: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 67259294123: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 67259294347: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 67333942395: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 67402564288: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 67460352292: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 67457479720: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 67509699623: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 67721426870: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 67717788339: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 67735168616: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 67825975117: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 67825978847: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 67866630653: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 67952765911: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 67952770259: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 67964360170: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 67964360806: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 68045986950: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 68045994560: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 68045994834: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 68045995048: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 68063598191: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 68222264957: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 68338560199: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 68380368284: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 68380368332: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 68380373100: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 68374444206: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 68374444461: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 68771868179: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 68771870158: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 68771872500: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 68771875667: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 68771883898: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 68846953296: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 68848346759: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 68905201002: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 68905214723: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 68905215337: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 68905216710: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 68943877994: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 68944654889: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 68992131789: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 69004321773: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 69012990364: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 69036263974: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 69030888068: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 69035835620: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 69057430328: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 69085666645: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 69085365125: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 69085365173: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 69085368602: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 69135874782: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 69138011987: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 69138012113: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 8: Match found at file offset 69138013862: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 69138022890: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 69138023795: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 69138034185: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 69138036019: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 69138037639: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 69138039737: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 69138041471: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 69153464424: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 69161388289: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 69167035821: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 69170854821: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69196448728: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69196448859: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69196451954: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69196452009: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69196458432: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69263507291: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297795663: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297796106: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297798367: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297802825: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297804759: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297805199: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297807613: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297810032: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69297855480: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69298909976: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69315068137: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315035102: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315035269: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315035424: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315035534: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315037285: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315044659: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315047634: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315050525: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315051017: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315062873: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315063274: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315065247: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315066270: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315066645: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 69315067669: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 69338453731: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 69385799488: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 69372726634: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 69493564286: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 69554751562: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 69554751732: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69737741053: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 69829009618: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 69852993870: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 69853000489: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 69870322797: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 69902784170: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 69927113769: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 69927115821: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 69963275767: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 69963276632: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 69963289462: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 69972568991: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 69972637100: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 69962535155: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 69962536718: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 69962537321: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 69962539075: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 69962539400: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972409293: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972410718: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972418002: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972418053: ‘ Sylvester Stalone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972418054: ‘Sylvester Stalone ’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972418510: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972423248: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972454274: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972494855: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 69972509561: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 70194668464: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 70268141449: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 70268141835: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 70502320131: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 70502320259: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 70502320561: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 70502320854: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 70547668138: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 70562145394: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 70742642153: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 70853870365: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 6: Match found at file offset 70853873289: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 70879732827: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 70895251970: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 70964916826: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 71008682417: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 71145288115: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 71165145487: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 71188496073: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 71196630569: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 71393920615: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 71444754682: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 71468875095: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 71468875302: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 71566059101: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 71566060891: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 71566079166: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 71566112875: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 71566222989: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 71566253339: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 71611096223: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 71619261851: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 71742804099: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 72053680424: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 72068222004: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 72068227180: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 72068228918: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 72361254101: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 72371261153: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 72371265930: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 72371267675: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 72371268193: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 72387355732: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 72387357206: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 72387358861: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 72403434749: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 72785316311: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 72817182238: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 72817196717: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 72895189789: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 72906284257: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 73086122857: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 73086742084: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 73145430497: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 73220600976: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 73266390864: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 73300492301: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 73314775538: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 73438871673: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 73582015742: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 73636942502: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 73636961984: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 73637256763: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 73662708335: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 73662764848: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 73662765198: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 73662765434: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 73712461178: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 73719721993: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 73744829337: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 73817144237: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 73991808801: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 73991809078: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 73991809318: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 74024244459: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 74024246652: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 74017136872: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 74024191873: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 74024192740: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 74071737564: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74271763583: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 74298160618: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 74327250359: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 74559129527: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 74559131030: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 74559193195: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 74558925167: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 74599539954: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 74629379409: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 74728186529: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756760780: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756762392: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756768026: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756769331: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756769479: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756770350: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756770822: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756771636: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756773373: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 74756774822: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 8: Match found at file offset 74823523935: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 74823524392: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 74851148313: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 74947891816: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 74947892155: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 74965416546: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 74965417064: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 75060569633: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 75095297144: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 75095313542: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 75133432937: ‘[Sylvester Stalone’ (Edit Distance: 4)
Thread 5: Match found at file offset 75133432938: ‘Sylvester Stalone]’ (Edit Distance: 4)
Thread 6: Match found at file offset 75347604974: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 75370121107: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 75406525282: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 75406525602: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 75470759291: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 75533407118: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 75951958569: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 75959141840: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 75959146457: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 75983248957: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 75983249337: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 75983249421: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 75983249749: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 76066921274: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 76078430982: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 76078431212: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 76078435446: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 76078442150: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 76078476491: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 76082848597: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 76082849671: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 76082850451: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 76082855368: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 76082857723: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 76082859237: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 76132635718: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 76171302491: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 76171302705: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 76162044611: ‘Sylvestor Stallone’ (Edit Distance: 3)
Thread 8: Match found at file offset 76178180244: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 76178181871: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 76178183329: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 76204088657: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 76365504239: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 76425627279: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 76445026405: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 76445049805: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 76525688855: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 76606955293: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 76606955648: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 76606844674: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 76741469376: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 76741469968: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 76956988111: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 76992227926: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 77081118523: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 77081134402: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 77156072123: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 77193883093: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 77315229263: ‘ Sylvester Stalone’ (Edit Distance: 4)
Thread 11: Match found at file offset 77315229264: ‘Sylvester Stalone,’ (Edit Distance: 4)
Thread 10: Match found at file offset 77399848791: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 77423839359: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 77677227662: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 77681241239: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 77681241960: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 77772461771: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 77772462416: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 77772462795: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 77782958820: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 77798311000: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 77968034698: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 78109018302: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 78109018384: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 78109020390: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 78115605085: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 78124628316: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 78214724185: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 78246809168: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 78246823237: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 78305902410: ‘Sylvestor Stallone’ (Edit Distance: 3)
Thread 11: Match found at file offset 78305903222: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 78396436853: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 78588378324: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 78831356768: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 78896775453: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 78934509004: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 10: Match found at file offset 78951423299: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 79054387150: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 79054390917: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 79083327473: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 79083328345: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 79350623199: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 79350624850: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 79350628146: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 79350635426: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 79421853761: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 79567918677: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 79742583535: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 79848118188: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 79860724444: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 79890394823: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 80170458065: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 80170458171: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 80670591377: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 80715035897: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 80715036085: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 80880751183: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 80880751241: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 80880751486: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 80912829117: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 81020433529: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 81136817224: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 81344815477: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 81393937935: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 81448890641: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 81515378909: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 81515381056: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 81515382450: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514041768: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514043341: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514043656: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514044439: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514044626: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514046222: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514046955: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514047932: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514048218: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 81514051444: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 81805869311: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 81922426787: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 81922427172: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 81922427423: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 82379645659: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 82379650352: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 82379650594: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 82379651245: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 82412719235: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 82458891541: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 82484435724: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 82484436510: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 82484436544: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 82506705391: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 82506719323: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 82530762729: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 82530767621: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 82530767904: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 82555107694: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 82614221701: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 82702692839: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 82748598492: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 82822762471: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 82822762887: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 83115586271: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 83224260893: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670171358: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670172696: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670173304: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670174522: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670175073: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670176676: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670286970: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670287390: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670287598: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670287640: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670287882: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670288436: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670289358: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 83670289912: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 83690077668: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 83690117274: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 83690117325: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 83690117535: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 83753247244: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 84072579475: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 84175314927: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 8: Match found at file offset 84175406355: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 84202066945: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 84202069408: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 84215303210: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 84215327936: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 84274913401: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 84313638173: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 84315506467: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 84326710585: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 84355044042: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 84371301770: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 84371309810: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 84397477666: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 84524858608: ‘Silvester Stallone’ (Edit Distance: 3)
Thread 13: Match found at file offset 84546037673: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 84546038284: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 84546039035: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 84588423153: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 84588424137: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 84588430985: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 84765798205: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 84970105119: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 84983788205: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 84983788824: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036478389: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036478793: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036479584: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036480195: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036480643: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036485633: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036486224: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036486325: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036486676: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036487605: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036488046: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036488511: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036488976: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036489878: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036490405: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036491544: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036492102: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036493036: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036494242: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 85036496857: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 85113835440: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 85218881783: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85213347938: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 85309043248: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 85309417756: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 85373398460: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559110773: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559136094: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559141199: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559154933: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559179420: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559185821: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559227467: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559231725: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85559236205: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85678681391: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85678685483: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 85678686096: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 85727932938: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 85727933129: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 85727950592: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 85807842278: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 85868301901: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 85959210175: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 85986847638: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 85977680771: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 86034932341: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 86034932505: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 86034941747: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 86034945246: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 86034945394: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 86034980545: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 86034993940: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 86034994686: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 86070972980: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 86336101991: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 86398234659: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483832691: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483832903: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483835348: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483836325: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483844371: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483844978: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483849045: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483852725: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 15: Match found at file offset 86483853382: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86483853845: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 86782129291: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 86967137082: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 86967137191: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87060789405: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 87172341815: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 87164357913: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 87164358123: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 87164359987: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 87164391113: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 87164461148: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87228616974: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311183092: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311183167: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311183402: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311183565: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311184518: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311184900: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311185882: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311186198: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311187496: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311188702: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311189319: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311190012: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311190398: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311191014: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311191409: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311192079: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311192701: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311193930: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311195678: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 87311197408: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87350596161: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87350596291: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87350597135: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87350597245: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87350597669: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87350598120: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87350598554: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87432810878: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 87497429395: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 87508305951: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 87508306429: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 87698672011: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 87754354089: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 87857234378: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 87882384960: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 87882388065: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 87942796719: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 87942797259: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 87942834478: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 87942835999: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 87942838648: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 88043754056: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 88043755968: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 88045326432: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 88045511773: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 88206505258: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 88278420897: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 88278421147: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 88278421171: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 88278425272: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 88278425295: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 88618722800: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 88618723552: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 88618725984: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 88853314419: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 88926710152: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 88926712982: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 89065143076: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 89077782156: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 89295043856: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 89401804894: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 89435891198: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 89435922820: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89688779022: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 89759406549: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 89759408512: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 89759409148: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 89759409653: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 89759410480: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 89759410679: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 89753408836: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 89846733689: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883570792: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883571002: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883571282: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883571599: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 8: Match found at file offset 89883573149: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883579194: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883580186: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883580584: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883580893: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883581891: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 89883583413: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 89895804261: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 89946148346: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 89979744138: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 90002940806: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 90003046643: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 90005386268: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 90417554491: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 90417554643: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 90540755663: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 90540756051: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 90570461091: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 90872827645: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 90884051768: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 90951990835: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 91000782027: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146748057: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146750359: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146757797: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146758110: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146758677: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146759416: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146760998: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146761936: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146762591: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146765090: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146765599: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146766957: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146768844: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146775821: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146776593: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146777260: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146789507: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146791920: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 91146802173: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 91470449021: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 91495412873: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 91557292616: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 91642094624: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 91807534325: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 92058726032: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 92058799932: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 92058828486: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 92121843085: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 92123486230: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 92465580588: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 5: Match found at file offset 92479733078: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 92544606701: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 92553231530: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 10: Match found at file offset 92586470686: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 92910533870: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 93006608543: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 93006612208: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 93006612405: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 93039258081: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 4: Match found at file offset 93073023096: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290442798: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290443562: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290443725: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290454706: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290455110: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290489821: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290496824: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290497034: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 93290498495: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 93669851277: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 93697323559: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 93789221642: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 93832137563: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 93832140003: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 94078220915: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 94078221149: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 94111597066: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 94166327794: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 94189107101: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 94344755000: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 1: Match found at file offset 94391829234: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 94428175713: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 94428176595: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 94754102597: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 94892057523: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 94978677888: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 95062836417: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 1: Match found at file offset 95192184190: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95266560851: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 11: Match found at file offset 95436193595: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462361541: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462363583: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462363819: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462366211: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462368804: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462370503: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462372384: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462378694: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462382631: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462383464: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462384066: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462384745: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462386435: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462387589: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462389888: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462390814: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462392179: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462392991: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462395043: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 95462397197: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462288642: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462289482: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462290325: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462293331: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462310065: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462317894: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462318524: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462320021: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462320451: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462320879: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462321630: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462324405: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462330708: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462330906: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462330971: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462331071: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462331133: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462331687: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462332331: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462333423: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462334031: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462334560: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462335199: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462336429: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462337034: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462337751: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462339113: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462339705: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462340595: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462342396: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462343263: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462344230: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462344728: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462345488: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462346130: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462346828: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462347590: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462348199: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462348905: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462350519: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462351127: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462352084: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462352711: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462355661: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462356352: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 2: Match found at file offset 95462357575: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 95582136777: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 95839050480: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 95939428065: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 96039591854: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 96121446131: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 96181217672: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 14: Match found at file offset 96227971244: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 96288804599: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 96288805035: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 96288805644: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 6: Match found at file offset 96288806094: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 96338036046: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 96338037980: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 96338041675: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 7: Match found at file offset 96338049028: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 96492645327: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 96560820059: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 96560842563: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 0: Match found at file offset 96603626322: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 9: Match found at file offset 96648676494: ‘Sylvester Stallone’ (Edit Distance: 4)
```

```
Thread 1: Match found at file offset 96665975548: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 96685007516: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 96870624173: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 96870624246: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 96870624461: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 96870624759: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 96919770592: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 97224806290: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 97302858312: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 97302885946: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 97486187538: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 97517772714: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 97654980416: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 97654980435: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 97693619354: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 97780928238: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 97799216287: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 97799217921: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 97799240713: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 97829713339: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 98104663370: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 98138089426: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 98255821419: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 98255822053: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 98255824213: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 98250104607: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 98254478538: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 98254608576: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 98285191438: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 98310077457: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 98406492576: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 98540964478: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 98684797101: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 98707650057: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 98733248997: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 98890459320: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 98897019179: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 98958594878: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 98958595121: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 98958596177: 'Sylvester Stallone' (Edit Distance: 4)
Thread 8: Match found at file offset 99063365116: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 99056603648: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 99056603813: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 99056604156: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 99206773916: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 99206775215: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 99206776265: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 99206776368: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 99206782313: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 99276715810: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 99319030241: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 99319030528: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 99319030857: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 99400292692: 'Silvester Stallone' (Edit Distance: 3)
Thread 5: Match found at file offset 99584489366: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 99584490118: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 99584490428: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 99655877947: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 99681747349: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 99697072461: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 99851937699: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 99868804133: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 99868811171: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 99868811636: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 99925052129: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 99925056760: 'Sylvestor Stallone' (Edit Distance: 3)
Thread 11: Match found at file offset 99925057112: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 100001753294: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 100001753499: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 100098689955: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 100094620246: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100193060037: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 100377140333: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 100377140364: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 100529065118: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 100586638045: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 100586638100: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 100586640632: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655032343: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655032831: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655033031: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655033395: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655034093: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655034373: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655035178: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655035533: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655035906: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655036533: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 100655036652: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 0: Match found at file offset 100657968638: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 100788033317: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 100839404433: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 100867894155: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 100867894268: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 100867894865: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 100867894916: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 100881262091: 'Sylvester Stallone' (Edit Distance: 4)
Thread 14: Match found at file offset 100885574998: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 100916500028: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101093551571: 'Sylvester Stallone' (Edit Distance: 4)
Thread 5: Match found at file offset 101171390903: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259982431: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259982644: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259983539: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259984329: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259984541: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259985214: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259985906: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259987177: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259988757: 'Sylvester Stallone' (Edit Distance: 4)
Thread 0: Match found at file offset 101259989389: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 101262752243: 'Sylvester Stallone' (Edit Distance: 4)
Thread 7: Match found at file offset 101262752415: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 101306248623: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 101325295994: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 101325296304: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355640324: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355640809: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355640845: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355641160: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355641494: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355641813: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355642193: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355642507: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 101355642960: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 101410060198: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 101410060350: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 101410060971: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 101410061060: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 101410061355: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 101410061867: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 101410062955: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 101611496922: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 101783409402: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 101938032608: 'Sylvester Stallone' (Edit Distance: 4)
Thread 12: Match found at file offset 102065317921: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102095265437: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128399517: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128400119: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128400295: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128400584: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128400892: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128401180: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128401517: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128401895: 'Sylvester Stallone' (Edit Distance: 4)
Thread 9: Match found at file offset 102128402386: 'Sylvester Stallone' (Edit Distance: 4)
Thread 6: Match found at file offset 102446216854: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 102447442830: 'Sylvester Stallone' (Edit Distance: 4)
Thread 1: Match found at file offset 102447443502: 'Sylvester Stallone' (Edit Distance: 4)
Thread 11: Match found at file offset 102505129940: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 102623055010: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 102623055213: 'Sylvester Stallone' (Edit Distance: 4)
Thread 4: Match found at file offset 102637942066: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 102664825756: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 102767793782: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 102767794119: 'Sylvester Stallone' (Edit Distance: 4)
Thread 13: Match found at file offset 102767794484: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 102862464680: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 102892940566: 'Sylvester Stallone' (Edit Distance: 4)
Thread 10: Match found at file offset 102892941246: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 103113003837: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 103145810302: 'Sylvester Stallone' (Edit Distance: 4)
Thread 15: Match found at file offset 103145815283: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103146978700: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103146978747: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147035059: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147035091: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147191191: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147242101: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147242521: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147242593: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147243290: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147243407: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147243499: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103147243884: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103428037152: 'Sylvester Stallone' (Edit Distance: 4)
Thread 2: Match found at file offset 103428037305: 'Sylvester Stallone' (Edit Distance: 4)
Thread 3: Match found at file offset 103472106798: 'Sylvester Stallone' (Edit Distance: 4)
```

```
Thread 15: Match found at file offset 103491430556: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 103718367581: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 103759572101: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 103836310733: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 8: Match found at file offset 104014651634: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 13: Match found at file offset 104020825793: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 12: Match found at file offset 104172906983: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 3: Match found at file offset 104170025495: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 104187580126: ‘Sylvester Stallone’ (Edit Distance: 4)
Thread 15: Match found at file offset 104187580310: ‘Sylvester Stallone’ (Edit Distance: 4)

 Performance counter stats for './DraFF.elf enwiki-20241001-pages-articles.xml Silvestor Staloune 4':

     73,106,076.00 msec task-clock                       #   15.855 CPUs utilized
         1,523,775      context-switches                 #   20.843 /sec
            46,362      cpu-migrations                   #    0.634 /sec
             4,237      page-faults                      #    0.058 /sec
262,471,994,236,758      cycles                          #    3.590 GHz                      (71.43%)
  4,504,501,440,625      stalled-cycles-frontend         #    1.72% frontend cycles idle     (71.43%)
658,462,260,348,721      instructions                    #    2.51  insn per cycle
                                                         #    0.01  stalled cycles per insn  (71.43%)
 50,407,900,737,078      branches                        #  689.517 M/sec                    (71.43%)
     50,271,076,298      branch-misses                   #    0.10% of all branches          (71.43%)
187,888,060,698,697      L1-dcache-loads                 #    2.570 G/sec                    (71.43%)
     26,142,305,647      L1-dcache-load-misses           #    0.01% of all L1-dcache accesses  (71.43%)
    <not supported>      LLC-loads
    <not supported>      LLC-load-misses

    4610.935087820 seconds time elapsed

   72302.060321000 seconds user
     109.823665000 seconds sys

[root@djudjeto SS_vs_enwiki]#

[sanmayce@djudjeto SS_vs_enwiki]$ sort -u enwiki-20241001-pages-articles.xml.hits -o enwiki-20241001-pages-articles.xml.hits.unique
[sanmayce@djudjeto SS_vs_enwiki]$ cat enwiki-20241001-pages-articles.xml.hits.unique
Salvestor Stallone
Silvester Stallone
=Silvester Stalone
Silvester Stalone
Silvestro Stallone
:Silvestr Stallone
Silvestr Stallone]
Sylvester Stallone
 Sylvester Stalone
>Sylvester Stalone
[Sylvester Stalone
¦Sylvester Stalone
Sylvester Stalone
Sylvester Stalone,
Sylvester Stalone.
Sylvester Stalone<
Sylvester Stalone]
Sylvester Stalone¦
Sylvestor Stallone
[sanmayce@djudjeto SS_vs_enwiki]$
```

Kazahana.txt (in fact 'Kazahana.txt' numbered with $ nl Kazahana.txt>nlKazahana.txt):

1 In 1984, Cameron was hired to write a sequel to ''[[First Blood]]''; it was rewritten by [[Sylvester Stallone]] and released as ''[[Rambo: First Blood Part II]]''.&lt;ref&gt;{{Cite web |last=Crocker |first=Jonathan |date=November 19, 2009 |title=James Cameron Interview: On His Own Movies |url=https://www.totalfilm.com/features/james-cameron-interview-on-his-own-movies/rambo-first-blood-part-ii-1985 |archive-url=https://web.archive.org/web/20130722153924/https://www.totalfilm.com/features/james-cameron-interview-on-his-own-movies/rambo-first-blood-part-ii-1985 |archive-date=July 22, 2013 |access-date=January 21, 2023 |website=Total Film}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=https://www.nytimes.com/1985/05/22/movies/screen-sylvester-stallone-returns-as-rambo.html|title=Screen: Sylvester Stallone Returns as Rambo|last=Canby|first=Vincent|date=May 22, 1985|work=The New York Times|access-date=October 23, 2019|archive-url=https://web.archive.org/web/20190920055754/https://www.nytimes.com/1985/05/22/movies/screen-sylvester-stallone-returns-as-rambo.html|archive-date=September 20, 2019|language=en-US|issn=0362-4331}}&lt;/ref&gt; Cameron was then hired to write and direct a sequel to ''[[Alien (film)|Alien]]'' (1979), a science fiction horror film directed by [[Ridley Scott]]. Like the original, the sequel ''[[Aliens (film)|Aliens]]'' (1986) featured [[Sigourney Weaver]] as [[Ellen Ripley]]. ''Aliens'' follows Ripley as she helps a group of marines fight off [[Extraterrestrial life|extraterrestrials]]. Despite conflicts with cast and crew during production, and having to replace one of the lead actors – [[James Remar]] with [[Michael Biehn]] – ''Aliens'' was a box office success, generating over $130&amp;nbsp;million worldwide.&lt;ref&gt;{{Cite web|url=https://www.boxofficemojo.com/release/rl3309340161/weekend/|title=Aliens|website=Box Office Mojo|access-date=October 24, 2019|archive-url=https://web.archive.org/web/20191024170514/https://www.boxofficemojo.com/release/rl3309340161/weekend/|archive-date=October 24, 2019|url-status=live}}&lt;/ref&gt; The film was nominated for seven [[Academy Awards]] in 1987; [[Academy Award for Best Actress|Best Actress]], [[Academy Award for Best Production Design|Best Art Direction]], [[Academy Award for Best Film Editing|Best Film Editing]], [[Academy Award for Best Original Score|Best Original Score]] and [[Academy Award for Best Sound Mixing|Best Sound]]. It won awards for [[Academy Award for Best Sound Editing|Best Sound Editing]] and [[Academy Award for Best Visual Effects|Best Visual Effects]].&lt;ref&gt;{{Cite web|url=https://www.oscars.org/oscars/ceremonies/1987|title=The 59th Academy Awards {{!}} 1987|website=Oscars.org {{!}} Academy of Motion Picture Arts and Sciences|language=en|access-date=April 10, 2020|archive-url=https://web.archive.org/web/20170102081445/https://www.oscars.org/oscars/ceremonies/1987|archive-date=January 2, 2017|url-status=live}}&lt;/ref&gt; In addition, Weaver and the film made the cover of ''[[Time (magazine)|Time]]'' in July 1986.&lt;ref&gt;{{Cite web|url=http://www.tcm.com/thismonth/article/?cid=87922|title=Aliens|date=December 16, 2018|website=Turner Classic Movies|url-status=dead|archive-url=https://web.archive.org/web/20181216163340/http://www.tcm.com/this-month/article/87922%7C0/Aliens.html|archive-date=December 16, 2018|access-date=October 24, 2019}}&lt;/ref&gt;

2 Crosby won the [[Academy Award for Best Actor]] for his performance in ''[[Going My Way]]'' (1944) and was nominated for its sequel, ''[[The Bells of St. Mary's]]'' (1945), opposite [[Ingrid Bergman]], becoming the first of six actors to be nominated twice for playing the same character. Crosby was the number one box office attraction for five consecutive years from 1944 to 1948.&lt;ref name=Stanley&gt;Stanley, Bob, ''Let's Do It: The Birth of Pop Music'', Pegasus Books, 2022, pg. 220&lt;/ref&gt; At his screen apex in 1946, Crosby starred in three of the year's five highest-grossing films: ''The Bells of St. Mary's'', ''[[Blue Skies (1946 film)|Blue Skies]]'', and ''[[Road to Utopia]]''.&lt;ref name=Stanley/&gt; In 1963, he received the first [[Grammy Award|Grammy Global Achievement Award]].&lt;ref name=&quot;variety&quot;&gt;{{cite news |last=Tapley |first=Krostopher |title=Sylvester Stallone Could Join Exclusive Oscar Company with 'Creed' Nomination |work=[[Variety (magazine)|Variety]] |date=December 10, 2015 |url=https://variety.com/2015/film/in-contention/sylvester-stallone-oscars-golden-globes-1201657048/ |access-date=February 29, 2016}}&lt;/ref&gt; Crosby is one of 33 people to have three stars on the [[Hollywood Walk of Fame]],&lt;ref&gt;{{cite web |url=https://projects.latimes.com/hollywood/star-walk/about/#most-stars |title=Hollywood Star Walk |website=Projects.latimes.com}}&lt;/ref&gt; in the categories of motion pictures, radio, and audio recording.&lt;ref&gt;{{cite web |url=https://walkoffame.com/bing-crosby/ |title=Bing Crosby |website=Hollywood Walk of Fame|date=October 25, 2019 }}&lt;/ref&gt; He was also known for his collaborations with his friend [[Bob Hope]], starring in the ''[[Road to ...]]'' films from 1940 to 1962.

3 In the &quot;[[sound film|talkie]]&quot; era, Cleveland featured in several [[Major film studio|major studio]] films, such as [[Michael Curtiz]]'s [[Pre-Code Hollywood|pre-Code]] classic ''[[Goodbye Again (1933 film)|Goodbye Again]]'' (1933) with [[Warren William]] and [[Joan Blondell]]. Players from the [[1948 Cleveland Indians season|1948 Cleveland Indians]] appeared in ''[[The Kid from Cleveland]]'' (1949). [[Billy Wilder]]'s ''[[The Fortune Cookie]]'' (1966) was set and filmed in the city and marked the first onscreen pairing of [[Walter Matthau]] and [[Jack Lemmon]]. Labor struggles in Cleveland were depicted in ''[[Native Land]]'' (1942), narrated by [[Paul Robeson]], and in [[Norman Jewison]]'s ''[[F.I.S.T. (film)|F.I.S.T.]]'' (1978) with [[Sylvester Stallone]]. Clevelander [[Jim Jarmusch]]'s ''[[Stranger Than Paradise]]'' (1984) - a [[deadpan]] comedy about two New Yorkers who travel to [[Florida]] by way of Cleveland - was a favorite of the [[Cannes Film Festival]]. ''[[Major League (film)|Major League]]'' (1989) reflected the [[Cleveland Guardians#1960-1993: The 33-year slump|perennial struggles]] of the Cleveland Indians, while ''[[American Splendor (film)|American Splendor]]'' (2003) reflected the life of Cleveland graphic novelist [[Harvey Pekar]]. ''[[Kill the Irishman]]'' (2011) depicted the 1970s turf war between [[Danny Greene]] and the Cleveland crime family.&lt;ref name=&quot;Cle-Enc-Film&quot; /&gt;

4 Judge Dredd made his live-action debut in 1995 in ''[[Judge Dredd (film)|Judge Dredd]]'', portrayed by [[Sylvester Stallone]]. Later, he was portrayed by [[Karl Urban]] in the 2012 adaptation ''[[Dredd]]''. In audio dramas by [[Big Finish Productions]], Dredd is voiced by [[Toby Longworth]].

5 An American [[Judge Dredd (film)|film]] loosely based on the comic strip was released in 1995, starring [[Sylvester Stallone]] as Dredd. The film received generally negative reviews. On [[Rotten Tomatoes]] it has a 22% rating, and the site's critical consensus states that &quot;Director [Danny] Cannon fails to find the necessary balance to make it work&quot;.&lt;ref&gt;{{cite web |url=https://www.rottentomatoes.com/m/judge_dredd/ |title=Judge Dredd |date=30 June 1995 |work=[[Rotten Tomatoes]] |access-date=2021-10-01 }}&lt;/ref&gt; In deference to its expensive Hollywood star, Dredd's face was shown. In the comic, he very rarely removes his helmet, and even then his real face is never revealed. Also, the writers largely omitted the ironic humour of the comic strip, and ignored important aspects of the &quot;Dredd mythology&quot;. The co-creator and main writer of the comic character, [[John Wagner]], said: {{blockquote|I hated that plot. It was Dredd pressed through the Hollywood cliché mill, a dynastic power struggle that had little connection with the character we know from the comic.&lt;ref name=&quot;2000adreview_John-Wagner-on-Dredd&quot;&gt;{{cite web |first=Gavin |last=Hanly |url=http://www.2000adreview.co.uk/site/index.php/Interviews/John-Wagner-on-Dredd.html |title=John Wagner on Dredd |publisher=2000 AD Review |date=19 January 2010 |archive-url = https://web.archive.org/web/20100201230126/http://www.2000adreview.co.uk/site/index.php/Interviews/John-Wagner-on-Dredd.html | archive-date = 1 February 2010 |access-date=27 August 2016}}&lt;/ref&gt; }}

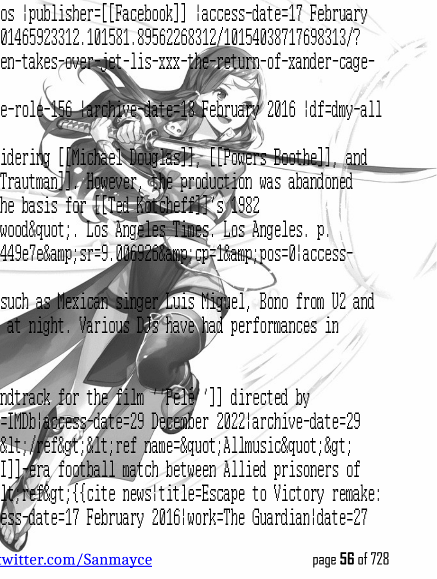
6 *[[Alain Dorval]] (1946-2024), French voice actor. He dubbed [[Sylvester Stallone]] in French versions.

7 Schwarzenegger's breakthrough film was the [[sword and sorcery]] epic ''[[Conan the Barbarian (1982 film)|Conan the Barbarian]]'' in 1982, which was a box-office hit.&lt;ref name=&quot;katzfilm&quot; /&gt; This was followed by a sequel, ''[[Conan the Destroyer]]'', in 1984, although it was not as successful as its predecessor.&lt;ref name=&quot;tessay&quot;&gt;{{cite news |first=Clark |last=Collis |title=Empire Essay: The Terminator |url=https://www.empireonline.com/reviews/reviewcomplete.asp?FID=132648 |archive-url=https://web.archive.org/web/20070927235924/http://www.empireonline.com/reviews/reviewcomplete.asp?FID=132648 |archive-date=September 27, 2007 |work=Empire |access-date=April 18, 2008}}&lt;/ref&gt; In 1983, Schwarzenegger starred in the promotional video ''[[Rio Carnival|Carnival]] in Rio''.&lt;ref&gt;{{cite magazine | url=https://ew.com/article/1992/06/12/carnival-rio/ | title=Carnival in Rio | magazine=[[Entertainment Weekly]] | first=Steve | last=Daly | date=June 12, 1992 | access-date=January 21, 2022 | archive-date=January 21, 2022 | archive-url=https://web.archive.org/web/20220121062453/https://ew.com/article/1992/06/12/carnival-rio/ | url-status=live }}&lt;/ref&gt; In 1984, he made his first appearance as the eponymous character in [[James Cameron]]'s science fiction action film ''[[The Terminator]]''.&lt;ref name=&quot;katzfilm&quot; /&gt;&lt;ref name=&quot;profilear&quot; /&gt;&lt;ref name=&quot;fanleam&quot; /&gt; It has been called his acting career's signature role.&lt;ref&gt;{{cite web | url= https://www.independent.co.uk/arts-entertainment/films/schwarzenegger-s-back-as-terminator-2275459.html | title= Schwarzenegger's back as Terminator | work= [[The Independent]] | date= April 27, 2011 | access-date= January 21, 2022 | archive-date= January 21, 2022 | archive-url= https://web.archive.org/web/20220121061722/https://www.independent.co.uk/arts-entertainment/films/schwarzenegger-s-back-as-terminator-2275459.html | url-status= live }}&lt;/ref&gt; Following this, Schwarzenegger made another sword and sorcery film, ''[[Red Sonja (1985 film)|Red Sonja]]'', in 1985.&lt;ref name=&quot;tessay&quot; /&gt; During the 1980s, audiences had an appetite for action films, with both Schwarzenegger and [[Sylvester Stallone]] becoming international stars.&lt;ref name=&quot;profilear&quot; /&gt; During the [[Schwarzenegger-Stallone rivalry]] they attacked each other in the press, and tried to surpass the other with more on-screen killings and larger weapons.{{r|pearson20171009}} Schwarzenegger's roles reflected his sense of humor, separating him from more serious action hero films. He made a number of successful action films in the 1980s, such as ''[[Commando (1985 film)|Commando]]'' (1985), ''[[Raw Deal (1986 film)|Raw Deal]]'' (1986), ''[[The Running Man (1987 film)|The Running Man]]'' (1987), ''[[Predator (film)|Predator]]'' (1987), and ''[[Red Heat (1988 film)|Red Heat]]'' (1988).

8 His film appearances after becoming Governor of California included a three-second cameo appearance in ''[[The Rundown]]'' and the 2004 remake of ''[[Around the World in 80 Days (2004 film)|Around the World in 80 Days]]''. In 2005, he appeared as himself in the film ''[[The Kid &amp; I]]''. He voiced [[Baron von Steuben]] in the ''[[Liberty's Kids]]'' episode &quot;[[Valley Forge]]&quot;. He had been rumored to be appearing in ''[[Terminator Salvation]]'' as the original T-800; he denied his involvement,&lt;ref&gt;{{cite web |url=http://scifiwire.com/2009/03/arnold-confirms-but-downp.php |title=Arnold downplays a Terminator Salvation cameo |publisher=SCI FI Wire |access-date=March 11, 2009 |archive-url=https://web.archive.org/web/20090313234133/http://scifiwire.com/2009/03/arnold-confirms-but-downp.php |archive-date=March 13, 2009 }}&lt;/ref&gt; but he ultimately did appear briefly via his image being inserted into the movie from stock footage of the first ''Terminator'' movie.&lt;ref&gt;{{cite web |url=http://movies.tvguide.com/Movie-News/Arnold-Schwarzenegger-Terminator-1005894.aspx |title=Arnold Schwarzenegger (Virtually) Back in Terminator Salvation |work=TV Guide |access-date=May 8, 2009 |archive-url=https://web.archive.org/web/20090520083012/http://movies.tvguide.com/Movie-News/Arnold-Schwarzenegger-Terminator-1005894.aspx |archive-date=May 20, 2009 |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://www.reelzchannel.com/person/176340/mcg |title=McG Talks Terminator Salvation |publisher=reelzchannel.com |access-date=May 11, 2009 |archive-url=https://web.archive.org/web/20090501190521/http://www.reelzchannel.com/person/176340/mcg |archive-date=May 1, 2009 }}&lt;/ref&gt; Schwarzenegger appeared in Sylvester Stallone's ''[[The Expendables (2010 film)|The Expendables]]'' (2010), where he made a [[cameo appearance]].

9 On May 20, 2011, Schwarzenegger's entertainment counsel announced that all film projects currently in development were being halted: &quot;Schwarzenegger is focusing on personal matters and is not willing to commit to any production schedules or timelines.&quot;&lt;ref&gt;{{cite web |url=http://www.people.com/people/article/0,,20496056,00.html |title=Arnold Schwarzenegger Halts All Acting Projects – Including Terminator |first=Sara |last=Hammel |work=People |access-date=October 18, 2011 |archive-url=https://web.archive.org/web/20110902054617/http://www.people.com/people/article/0,,20496056,00.html |archive-date=September 2, 2011 |url-status=live }}&lt;/ref&gt; On July 11, 2011, it was announced that Schwarzenegger was considering a comeback film, despite legal problems related to his divorce.&lt;ref&gt;{{cite magazine |last=Finke |first=Nikki |url=https://www.deadline.com/2011/07/arnold-schwarzenegger-books-last-stand-western-will-test-his-post-scandal-popularity/ |title=Arnold Books 'Last Stand': Studio To Test Schwarzenegger's Post-Scandal Popularity – |magazine=Deadline |access-date=October 9, 2011 |date=July 11, 2011 |archive-url=https://web.archive.org/web/20111019084025/http://www.deadline.com/2011/07/arnold-schwarzenegger-books-last-stand-western-will-test-his-post-scandal-popularity/ |archive-date=October 19, 2011 |url-status=live }}&lt;/ref&gt; He starred in ''[[The Expendables 2]]'' (2012) as Trench Mauser,&lt;ref&gt;{{cite magazine |last=Fleming |first=Mike |title='Expendables 2' Sets Action Dream Trio: Arnold Schwarzenegger, Bruce Willis Join Sly Stallone |url=https://www.deadline.com/2011/09/expendables-2-sets-action-dream-trio-arnold-schwarzenegger-bruce-willis-join-sly-stallone/ |magazine=Deadline |access-date=October 18, 2011 |date=September 6, 2011 |archive-url=https://web.archive.org/web/20121020044920/http://www.deadline.com/2011/09/expendables-2-sets-action-dream-trio-arnold-schwarzenegger-bruce-willis-join-sly-stallone/ |archive-date=October 20, 2012 |url-status=live }}&lt;/ref&gt; and starred in ''[[The Last Stand (2013 film)|The Last Stand]]'' (2013), his first leading role in 10 years, and ''[[Escape Plan (film)|Escape Plan]]'' (2013), his first co-starring role alongside Sylvester Stallone. He starred in ''[[Sabotage (2014 film)|Sabotage]]'', released in March 2014, and returned as Trench Mauser in ''[[The Expendables 3]]'', released in August 2014. He starred in the fifth Terminator film ''[[Terminator Genisys]]'' in 2015.&lt;ref name=&quot;katzfilm&quot; /&gt;&lt;ref name=&quot;profilear&quot; /&gt;&lt;ref name=&quot;fanleam&quot;&gt;{{cite book |last=Leamer |first=Laurence |title=Fantastic: The life of Arnold Schwarzenegger

```
   |year=2005 |publisher=St Martin's Press |isbn=978-0-312-33338-6 |url=https://archive.org/details/fantasticlifeofa00leam }}&lt;/ref&gt;&lt;ref&gt;[https://screenrant.com/arnold-schwarzenegger-terminator-5/ Arnold Schwarzenegger Confirmed for 'Terminator 5'] {{Webarchive|url=https://web.archive.org/web/20130807205307/http://screenrant.com/arnold-schwarzenegger-terminator-5/ |date=August 7, 2013 }}. Screenrant.com (January 22, 2013). Retrieved September 27, 2013.&lt;/ref&gt; He then planned to reprise his role as [[Conan the Barbarian]] in ''[[Conan the Barbarian (1982 film)#Sequels|The Legend of Conan]]'',&lt;ref name=&quot;abcnews.go.com&quot;&gt;{{cite web |url=https://abcnews.go.com/Entertainment/arnold-schwarzenegger-file-revised-divorce-papers-denying-spousal/story?id=14142207 |title=Arnold Schwarzenegger to File Revised Divorce Papers, Not Denying Spousal Support |work=ABC News |date=July 23, 2011 |access-date=October 9, 2011 |archive-url=https://web.archive.org/web/20110723223016/http://abcnews.go.com/Entertainment/arnold-schwarzenegger-file-revised-divorce-papers-denying-spousal/story?id=14142207 |archive-date=July 23, 2011 |url-status=live }}&lt;/ref&gt;&lt;ref&gt;Fleming, Mike (October 25, 2012). &quot;[https://www.deadline.com/2012/10/arnold-and-conan-the-barbarian-reunited-universal-reboots-action-franchise-with-schwarzenegger/ Arnold And 'Conan The Barbarian' Reunited: Universal Reboots Action Franchise With Schwarzenegger] {{Webarchive|url=https://web.archive.org/web/20121028195611/http://www.deadline.com/2012/10/arnold-and-conan-the-barbarian-reunited-universal-reboots-action-franchise-with-schwarzenegger/ |date=October 28, 2012 }}.&quot; Deadline.com. Retrieved October 30, 2012.&lt;/ref&gt; later renamed ''[[Conan the Barbarian (1982 film)#Sequels|Conan the Conqueror]].'' &lt;ref name=&quot;ew.com Schwarzenegger Conan Sequel January 28, 2016&quot;&gt;{{cite magazine |url=https://www.ew.com/article/2016/01/28/arnold-schwarzenegger-conan-sequel-title-plot |title=Arnold Schwarzenegger reveals new title, plot details about Conan sequel (by Jonathan Dornbush) |date=January 28, 2016 |magazine=[[Entertainment Weekly]] |access-date=August 21, 2016 |archive-url=https://web.archive.org/web/20160818062640/http://www.ew.com/article/2016/01/28/arnold-schwarzenegger-conan-sequel-title-plot |archive-date=August 18, 2016 |url-status=live }}&lt;/ref&gt; However, in April 2017, producer Chris Morgan stated that Universal had dropped the project, although there was a possibility of a TV show. The story of the film was supposed to be set 30 years after the first, with some inspiration from [[Clint Eastwood]]'s ''[[Unforgiven]]''.&lt;ref&gt;{{cite web |url=https://www.slashfilm.com/arnold-schwarzeneggers-the-legend-of-conan-may-not-happen-after-all/ |title=Arnold Schwarzenegger's 'The Legend of Conan' May Not Happen After All |publisher=slashfilm.com |date=April 6, 2017 |access-date=April 7, 2017 |archive-url=https://web.archive.org/web/20170407061650/http://www.slashfilm.com/arnold-schwarzeneggers-the-legend-of-conan-may-not-happen-after-all/ |archive-date=April 7, 2017 |url-status=live }}&lt;/ref&gt;
10 Schwarzenegger transferred profits from the mail-order business and his bodybuilding-competition winnings by rolling the proceeds into his first real estate investment: an apartment building he purchased for $10,000. Schwarzenegger made millions of dollars by investing in a variety of real estate holding companies both within the United States and around the world.&lt;ref&gt;{{cite news |title=Schwarzenegger reveals pumped-up finances |work=San Francisco Chronicle |url=http://www.sfgate.com/cgi-bin/article.cgi?f=/chronicle/archive/2003/08/10/ARNOLD.TMP |date=August 10, 2003 |access-date=April 18, 2008 |last=Williams |first=Lance |archive-url=https://web.archive.org/web/20080318192048/http://www.sfgate.com/cgi-bin/article.cgi?f=%2Fchronicle%2Farchive%2F2003%2F08%2F10%2FARNOLD.TMP |archive-date=March 18, 2008 |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite web |title=The Best Salesman in America? |publisher=Selling Power |url=https://www.sellingpower.com/2010/02/02/6660/the-best-salesman-in-america |access-date=December 29, 2021 |last=Fleschner |first=Malcolm |archive-url=https://web.archive.org/web/20080224112746/http://www.sellingpower.com/article/display.asp?aid=SP4182687 |archive-date=February 24, 2008}}&lt;/ref&gt; Schwarzenegger and fellow Hollywood veteran actor and industry adversary [[Sylvester Stallone]] [[Burying the hatchet|brought their long-storied industry rivalry to an end]] by both investing in the [[Planet Hollywood]]&lt;ref name=&quot;pearson20171009&quot;&gt;{{Cite web |last=Pearson |first=Ben |date=October 9, 2017 |title=Schwarzenegger Orchestrated a Legendary Hollywood Troll |url=https://www.slashfilm.com/arnold-schwarzenegger-beyond-fest/ |access-date=July 27, 2020 |website=/Film |language=en-US |archive-date=July 27, 2020 |archive-url=https://web.archive.org/web/20200727050548/https://www.slashfilm.com/arnold-schwarzenegger-beyond-fest/ |url-status=live }}&lt;/ref&gt; chain of international theme restaurants (modeled after the [[Hard Rock Cafe]]) along with [[Bruce Willis]] and [[Demi Moore]]. However, Schwarzenegger severed his financial ties with the chain in early 2000.&lt;ref name=&quot;planetbbc&quot;&gt;{{cite news |url=http://news.bbc.co.uk/2/hi/entertainment/619317.stm |title=Arnie's Planet Hollywood split |publisher=[[BBC]] |date=January 26, 2000 |access-date=April 1, 2012 |archive-url=https://web.archive.org/web/20120531075351/http://news.bbc.co.uk/2/hi/entertainment/619317.stm |archive-date=May 31, 2012 |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite news |url=https://www.latimes.com/archives/la-xpm-2000-jan-26-fi-57832-story.html |title=It's Hasta la Vista, Planet Hollywood |first=Paul |last=Lieberman |work=[[Los Angeles Times]] |date=January 26, 2000 |access-date=April 1, 2012 |url-status=live |archive-url=https://web.archive.org/web/20120514174355/http://articles.latimes.com/2000/jan/26/business/fi-57832 |archive-date=May 14, 2012 }}&lt;/ref&gt; Schwarzenegger remarked that the restaurant did not achieve the success that he had hoped for, claiming he wanted to focus his attention on &quot;new US global business ventures&quot; and his then-burgeoning acting career.&lt;ref name=&quot;planetbbc&quot; /&gt;
11 * 1946   &amp;ndash; [[Sylvester Stallone]], American actor, director, and screenwriter
12 In late 1981, Parton began filming her second film, the [[musical film]] ''[[The Best Little Whorehouse in Texas (film)|The Best Little Whorehouse in Texas]]'' (1982).&lt;ref name=&quot;Hall of Fame&quot;/&gt; The film earned her a second nomination for a Golden Globe Award for Best Actress{{snd}}Motion Picture Musical or Comedy.&lt;ref name=&quot;Golden Globes&quot;/&gt; The film was greeted with positive critical reviews and became a commercial success, earning over $69{{spaces}}million worldwide. After a two-year hiatus from films, Parton was teamed with [[Sylvester Stallone]] for ''[[Rhinestone (film)|Rhinestone]]'' (1984). A comedy film about a country music star's efforts to mould an unknown into a music sensation, the film was a critical and financial failure, making just over $21{{spaces}}million on a $28{{spaces}}million budget.
13 |[[Iraj Nazerian]]||[[Orson Welles]]&lt;br /&gt;[[Marlon Brando]]&lt;br /&gt;[[Lino Ventura]]&lt;br /&gt;[[Gene Hackman]]&lt;br /&gt;[[Charles Bronson]]&lt;br /&gt;[[Sylvester Stallone]]&lt;br /&gt;[[Rock Hudson]]
14 |[[Isao Sasaki]] || [[Sylvester Stallone]]&lt;br /&gt;[[Elvis Presley]]&lt;br /&gt;[[David Hasselhoff]] ||
15 |[[Michio Hazama]] || [[Roy Scheider]]&lt;br /&gt;[[Steve Martin]]&lt;br /&gt;[[Dean Martin]]&lt;br /&gt;[[Sylvester Stallone]] ||
16 | [[:ko:이정구(성우)|Lee Jung-Goo]] || [[Nicolas Cage]]&lt;br /&gt;[[Richard Gere]]&lt;br /&gt;[[Sylvester Stallone]]&lt;br /&gt;[[Bruce Willis]] ||
17 | [[Alain Dorval]]|| [[Sylvester Stallone]]&lt;br /&gt;[[Nick Nolte]] ||
18 | [[Thomas Danneberg]] || [[Arnold Schwarzenegger]]&lt;br /&gt;[[Sylvester Stallone]]&lt;br /&gt;[[Nick Nolte]] ||
19 |[[Robert De Niro]]&lt;br /&gt;[[Dustin Hoffman]]&lt;br /&gt;[[Sylvester Stallone]]&lt;br /&gt;[[Tomas Milian]]&lt;br /&gt;[[Al Pacino]]&lt;br /&gt;[[Peter Falk]]
20 |The Italian voice of Robert De Niro, Dustin Hoffman, and Sylvester Stallone in almost each of their films from 1969 to 1995. He also dubbed the most famous interpretations of Al Pacino. He was chosen by Tomas Milian himself for his dubbed voice.
21 [[File:Hefner 1978.jpg|thumb|right|Hefner at the premiere of [[Sylvester Stallone|Sylvester Stallone's]] movie ''[[F.I.S.T. (film)|F.I.S.T.]]'' in 1978]]
22 His first role in a non-Chinese film was as a villain in ''[[Lethal Weapon 4]]'' (1998), and his first leading role in a [[Hollywood film]] was as Han Sing in ''[[Romeo Must Die]]'' (2000). He has gone on to star in many international action films, including in French cinema with the [[Luc Besson]]-produced films ''[[Kiss of the Dragon]]'' (2001) and ''[[Unleashed (2005 film)|Unleashed]]'' (2005). He co-starred in ''[[The One (2001 film)|The One]]'' (2001) and ''[[War (2007 film)|War]]'' (2007) with [[Jason Statham]], ''[[The Forbidden Kingdom]]'' (2008) with [[Jackie Chan]], the first three of ''[[The Expendables (film series)|The Expendables]]'' films with [[Sylvester Stallone]], and as the title character villain in ''[[The Mummy: Tomb of the Dragon Emperor]]'' (2008). He continued to be active in Hong Kong cinema, appearing in ''[[The Warlords]]'' (2007), by which he became the highest paid actor in a Chinese-language movie, previously holding the record for his part in ''Hero''.&lt;ref&gt;{{cite web | url=https://www.digitalspy.com/movies/a80627/jet-li-breaks-chinese-film-salary-record/ | title=Jet Li breaks Chinese film salary record | website=[[Digital Spy]] | date=26 November 2007 }}&lt;/ref&gt;
23 After a one-year hiatus from filmmaking, Li returned to acting in 2010, portraying a mercenary in the film ''[[The Expendables (2010 film)|The Expendables]]'', teaming up with action stars [[Sylvester Stallone]], [[Jason Statham]], [[Dolph Lundgren]], [[Mickey Rourke]], [[Eric Roberts]], [[Stone Cold Steve Austin|Steve Austin]], [[Terry Crews]], and [[Randy Couture]]. It was the third time he had teamed up with Statham. In 2012, he reprised his role briefly in the sequel ''[[The Expendables 2]]'' and returned for the third film ''[[The Expendables 3]]'' in 2014. Li was initially stated to be appearing with [[Vin Diesel]] in ''[[XXX: Return of Xander Cage]]'', but according to a Facebook post by Diesel, Li was replaced by [[Donnie Yen]].&lt;ref&gt;{{cite web |url=http://twitchfilm.com/2016/01/tony-jaa-jet-li-and-deepika-padukone-join-vin-diesel-in-xxx-the-return-of-xander-cage.html |title=Tony Jaa, Jet Li and Deepika Padukone Join Vin Diesel In XXX: THE RETURN OF XANDER CAGE |publisher=Twitch |last=Brown |first=Todd |date=1 January 2016 |url-status=live |archive-url=https://web.archive.org/web/20160121193204/http://twitchfilm.com/2016/01/tony-jaa-jet-li-and-deepika-padukone-join-vin-diesel-in-xxx-the-return-of-xander-cage.html |archive-date=21 January 2016 |df=dmy-all}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://www.facebook.com/VinDiesel/photos/a.101465923312.101581.89562268312/10154038717698313/?type=3&amp;permPage=1 |title=Vin Diesel / photos |publisher=[[Facebook]] |access-date=17 February 2016 |date=12 February 2016 |url-status=live |archive-url=https://web.archive.org/web/20171116203008/https://www.facebook.com/VinDiesel/photos/a.101465923312.101581.89562268312/10154038717698313/?type=3&amp;permPage=1 |archive-date=16 November 2017 |df=dmy-all}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://www.joblo.com/movie-news/donnie-yen-takes-over-jet-lis-xxx-the-return-of-xander-cage-role-156 |title=Donnie Yen Takes Over Jet Li's ''XXX: The Return of Xander Cage'' Role |date=12 February 2016 |url-status=live |archive-url=https://web.archive.org/web/20160218082548/http://www.joblo.com/movie-news/donnie-yen-takes-over-jet-lis-xxx-the-return-of-xander-cage-role-156 |archive-date=18 February 2016 |df=dmy-all |access-date=12 February 2016}}&lt;/ref&gt;
24 In 1977, [[Carter DeHaven]] hired Frankenheimer to direct William Sackheim and [[Michael Kozoll]]'s screenplay for ''[[First Blood]]''. After considering [[Michael Douglas]], [[Powers Boothe]], and [[Nick Nolte]] for the role of [[John Rambo]] Frankenheimer cast [[Brad Davis (actor)|Brad Davis]]. He also cast [[George C. Scott]] as [[Colonel Trautman]]. However, the production was abandoned after [[Orion Pictures]] acquired its distributor [[Filmways]], and Sackheim and Kozoll's script would be rewritten by [[Sylvester Stallone]] as the basis for [[Ted Kotcheff]]'s 1982 film.&lt;ref&gt;Broeske, Pat H. (November 25, 1985). &quot;The Curious Evolution of John Rambo: How He Hacked His Way Through the Jungles of Hollywood&quot;. Los Angeles Times. Los Angeles. p. AB32.&lt;/ref&gt;&lt;ref&gt;{{Cite web|title=AFI{{!}}Catalog|url=https://catalog.afi.com/Film/56779-FIRST-BLOOD?sid=0506c692-2c7e-4866-aa90-000b66449e7e&amp;sr=9.006926&amp;cp=1&amp;pos=0|access-date=2021-06-11|website=catalog.afi.com}}&lt;/ref&gt;
25 For example, Baby 'O has been open to the national and international public since 1976 and different celebrities have visited their installations such as Mexican singer Luis Miguel, Bono from U2 and Sylvester Stallone. Another nightclub is Palladium, located in the Escénica Avenue, the location gives the nightclub a view of the Santa Lucia Bay at night. Various DJs have had performances in Palladium among them DVBBS, Tom Swoon, Nervo and Junkie KID.{{citation needed|date=December 2018}}
26 * [[Oscar (1991 film)|''Oscar'' (1991 film)]], starring Sylvester Stallone
27 Pelé published several autobiographies, starred in documentary films, and composed musical pieces, including [[Sérgio Mendes]]' [[Pelé (album)|soundtrack for the film ''Pelé'']] directed by [[François Reichenbach]] in 1977.&lt;ref&gt;{{cite web|url=https://www.imdb.com/title/tt13046456/?ref_=nm_flmg_t_10_dr|title=Pelé (1977)|publisher=IMDb|access-date=29 December 2022|archive-date=29 December 2022|archive-url=https://web.archive.org/web/20221229194005/https://www.imdb.com/title/tt13046456/?ref_=nm_flmg_t_10_dr|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Allmusic&quot;&gt;{{AllMusic|class=album|id=pele-mw0000375622|label=Pelé}}&lt;/ref&gt; He appeared in the 1981 film ''[[Escape to Victory]]'', about a [[World War II]]-era football match between Allied prisoners of war and a German team. Pelé starred alongside other footballers of the 1960s and 1970s, with actors [[Michael Caine]] and [[Sylvester Stallone]].&lt;ref&gt;{{cite news|title=Escape to Victory remake: who should follow in Pelé's footsteps?|url=https://www.theguardian.com/film/filmblog/2014/jun/27/escape-to-victory-remake-film-who-should-star|access-date=17 February 2016|work=The Guardian|date=27
```



June 2014|archive-date=4 January 2016|archive-url=https://web.archive.org/web/20160104035259/http://www.theguardian.com/film/filmblog/2014/jun/27/escape-to-victory-remake-film-who-should-star|url-status=live}}&lt;/ref&gt; In 1969, Pelé starred in a telenovela called ''Os Estranhos'', about first contact with aliens. It was created to drum up interest in the [[List of Apollo missions|Apollo missions]].&lt;ref&gt;{{cite web|title=Estranhos extraterrestres chegavam na tela da Excelsior para fazer contato com Pelé|url=http://cartaodevisita.r7.com/conteudo/1257/estranhos-extraterrestres-chegavam-na-tela-da-excelsior-para-fazer-contato-com-pele|publisher=Cartâo de Visita|date=16 January 2013|access-date=17 February 2016|language=pt|archive-date=6 April 2016|archive-url=https://web.archive.org/web/20160406133449/http://cartaodevisita.r7.com/conteudo/1257/estranhos-extraterrestres-chegavam-na-tela-da-excelsior-para-fazer-contato-com-pele|url-status=live}}&lt;/ref&gt; In 2001, he had a cameo role in the football satire film ''[[Mike Bassett: England Manager]]''.&lt;ref&gt;{{cite news|title=Mike Bassett: England Manager (2001) | url=https://www.nytimes.com/movies/movie/255633/Mike-Bassett-England-Manager/overview|access-date=17 February 2016|url-status=dead|archive-url=https://web.archive.org/web/20160301055650/https://www.nytimes.com/movies/movie/255633/Mike-Bassett-England-Manager/overview|department=Movies &amp; TV Dept. |work=[[The New York Times]]|author=Mark Deming|year=2016|archive-date=1 March 2016}}&lt;/ref&gt; Pelé was asked to participate in the 2006 ESPN documentary film ''[[Once in a Lifetime: The Extraordinary Story of the New York Cosmos]]'', but declined when the producers refused to pay his requested $100,000 fee.&lt;ref&gt;{{cite web |last1=Hirshey |first1=David |title=Pelé: When Soccer Ruled the USA |url=http://www.espn.com/espn/eticket/story?page=cosmos |publisher=ESPN | access-date=30 September 2020 |archive-date=14 December 2021 |archive-url=https://web.archive.org/web/20211214010907/http://www.espn.com/espn/eticket/story?page=cosmos |url-status=live }}&lt;/ref&gt;

28 For the role of Kyle Reese, Orion wanted a star whose popularity was rising in the United States but who also would have foreign appeal. Orion co-founder [[Mike Medavoy]] had met [[Arnold Schwarzenegger]] and sent his agent the script for ''The Terminator''.{{sfn|Keegan|2009|p=39}} Cameron was uncertain about casting Schwarzenegger as Reese as he felt he would need someone even more famous to play the Terminator. [[Sylvester Stallone]] and [[Mel Gibson]] both turned down the Terminator role.&lt;ref&gt;{{cite web|url=http://uk.complex.com/pop-culture/2013/06/a-history-of-iconic-roles-that-famous-actors-turned-down/sylvester-stallone-as-the-terminator|title=A History of Iconic Roles That Famous Actors Turned Down - Sylvester Stallone as T-800 (''The Terminator'', 1984)| website=Complex|access-date=August 16, 2017|archive-date=August 13, 2017|archive-url=https://web.archive.org/web/20170813235852/http://uk.complex.com/pop-culture/2013/06/a-history-of-iconic-roles-that-famous-actors-turned-down/sylvester-stallone-as-the-terminator|url-status=live}}&lt;/ref&gt; Medavoy suggested [[O. J. Simpson]] but Cameron did not feel that Simpson, at that time, would be believable as a killer.&lt;ref name=&quot;OV-Hurd&quot;&gt;{{cite AV media|title=Other Voices documentary|date=2001|publisher=[[MGM Home Entertainment]]|people=Hurd, Gale Anne (producer)|work=The Terminator [Special Edition]|medium=DVD}}&lt;/ref&gt;{{sfn|Keegan|2009|p=40}}&lt;ref&gt;{{Cite magazine|last=McGovern|first=Joe|url=https://ew.com/article/2014/07/17/the-terminator-oral-history/| title='The Terminator' at 30: An oral history|date=July 17, 2014|magazine=[[Entertainment Weekly]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Yamato |first=Jen |url=https://www.latimes.com/entertainment-arts/movies/story/2019-10-25/o-j-simpson-terminator-james-cameron-arnold-schwarzenegger |title=James Cameron debunks that O.J. Simpson 'Terminator' painting myth: 'Arnold is wrong' |date=October 25, 2019 |work=Los Angeles Times}}&lt;/ref&gt;

29 Hewitt has a nickname, &quot;Rusty&quot;, which was given to him by Darren Cahill who at the time thought Hewitt resembled the character Rusty, from the [[National Lampoon's Vacation (film series)| National Lampoon film series]].&lt;ref&gt;{{cite web |url=http://www.foxnews.com/sports/2012/01/20/aussie-hewitt-back-in-3rd-round-after-injuries/ |title=Aussie Hewitt back in 3rd round after injuries |date=19 January 2014 |publisher=Fox News Channel |access-date=19 January 2014}}&lt;/ref&gt; Hewitt has also been given the nickname 'Rocky' by fans, which originated from his shouts of &quot;C'mon Balboa&quot;, in reference to the character [[Rocky Balboa]] from the [[Sylvester Stallone]] film ''[[Rocky]]''. Hewitt has also been compared to the character.&lt;ref&gt;{{cite web | url=https://www.smh.com.au/sport/tennis/boxing-clever-rocky-pumps-up-australias-little-fella-20021204-gdfwv7.html |date=4 December 2002 |access-date=7 January 2019 |title=Boxing clever: Rocky pumps up Australia's 'little fella' |website=The Sydney Morning Herald |agency=Australian Associated Press}}&lt;/ref&gt;

30 While he was in his late twenties, Tureaud won two tough-man competitions consecutively.{{sfn|Mr. T|1985|p=223}} The first aired as &quot;Sunday Games&quot; on NBC-TV under the contest of &quot;America's Toughest Bouncer&quot; which included throwing a {{convert|150|lb|kg|adj=on}} [[Stunt performer|stuntman]], and breaking through a {{convert|4|in|cm|adj=on}} wooden door.{{sfn|Mr. T| 1985|p=224}} For the first event, Tureaud came in third place. For the end, two finalists squared off in a boxing ring for a two-minute round to declare the champion. Making it to the ring as a finalist, he had as his opponent a {{convert|280|lb|kg|adj=on}} [[Honolulu]] bouncer named Tutefano Tufi.{{sfn|Mr. T|1985|p=226}} Within twenty seconds &quot;Mr. T&quot; gave the six foot five competitor a bloody nose, and later a bloody mouth. He won the match and thus the competition.{{sfn|Mr. T|1985|p=227}} The second competition was aired under the new name &quot;Games People Play&quot; on NBC-TV. When interviewed by [[Bryant Gumbel]] before the final boxing match, Mr T. said, &quot;I just feel sorry for the guy who I have to box. I just feel real sorry for him.&quot; {{sfn|Mr. T|1985|p=234}} This fight was scheduled to last three rounds, but Mr. T finished it in less than 54 seconds. The line, &quot;I don't hate him but... I pity the fool&quot; in the movie ''[[Rocky III]]'' was written by [[Sylvester Stallone]], who is reputed to have been inspired by the interview.&lt;ref&gt;{{cite web |url=http://www.huffingtonpost.com/brian-cronin/did-ba-baracus-never-actually-say_b_7836116.html |title=Did B.A. Baracus Never Actually Say 'I Pity the Fool' on 'The A-Team'? |last=Cronin |first=Brian |date=July 24, 2015 |website=Huffington Post |access-date=August 16, 2016 |quote=Before the final match, Mr. T explained to the commentator for the event, Bryant Gumbel, that 'I just feel sorry for the guy who I have to box. I just feel real sorry for him.' Sylvester Stallone caught this second competition and was intrigued by Mr. T and that line in particular.}}&lt;/ref&gt;

31 In 1980, Mr. T was spotted by [[Sylvester Stallone]] while taking part in NBC's &quot;America's Toughest Bouncer&quot; competition, a segment of NBC's ''[[Games People Play (1980 TV series)|Games People Play]]''.&lt;ref&gt;{{cite web |url=http://www.starpulse.com/Actors/Mr._T/Biography/ |title=Biography of Mr. T |publisher=Starpulse.com |url-status=dead |archive-url=https://web.archive.org/web/20040622195143/http://starpulse.com/Actors/Mr._T/Biography/ |archive-date=June 22, 2004 |access-date=August 16, 2016}}&lt;/ref&gt; Although his role in ''[[Rocky III]]'' was originally intended as just a few lines, Mr. T was eventually cast as Clubber Lang, the primary antagonist. His catchphrase &quot;I pity the fool!&quot; comes from the film; when asked if he hates Rocky, Lang replies, &quot;No, I don't hate Balboa, but I pity the fool.&quot; He subsequently appeared in another boxing film, ''[[Penitentiary 2]]'',&lt;ref&gt;{{cite news| url=https://www.nytimes.com/1982/04/02/movies/movies-in-penitentiary-ii-too-sweet-gordon-gets-out.html|title=MOVIES: IN 'PENITENTIARY II,' TOO SWEET GORDON GETS OUT|last=Canby|first=Vincent| work=[[The New York Times]]|date=April 2, 1982|access-date=January 10, 2017}}&lt;/ref&gt; and on an episode of the Canadian [[sketch comedy]] series ''[[Bizarre (TV series)|Bizarre]]'', where he fights and eats [[Super Dave Osborne]],&lt;ref&gt;{{cite news|url=https://www.theglobeandmail.com/arts/super-dave-plans-comeback/article20390534/|title=Super Dave plans comeback|last=Szklarski| first=Cassandra|work=[[The Globe and Mail]]|date=March 31, 2009|access-date=January 10, 2017}}&lt;/ref&gt; before accepting a television series role on ''[[The A-Team]]''. He also appeared in an episode of ''[[Silver Spoons]]'', reprising his old role as bodyguard to the character Ricky Stratton (played by [[Ricky Schroder]]).

32 The [[Middle East]] has also participated in the genre with actors such as [[Youssef Mansour (actor)|Youssef Mansour]] who became famous in the 1990's for his Egyptian films that relied on martial arts.&lt;ref&gt;{{Cite web|url=https://elcinema.com/en/person/1029757/|title=Youssef Mansour - Actor Filmography، photos، Video|website=elCinema.com}}&lt;/ref&gt;&lt;ref name=&quot;auto3&quot;&gt; {{Cite web|url=http://www.mafhoum.com/press5/148S63.htm|title=Cairo Times|website=www.mafhoum.com}}&lt;/ref&gt;&lt;ref name=&quot;auto1&quot;&gt;{{Cite web|url=https://www.masrawy.com/arts/zoom/details/2023/9/30/2474934/%D8%A8%D8%B9%D8%AF-15-%D8%B9%D8%A7%D9%85-%D8%A7-%D8%B9%D9%84%D9%89-%D9%88%D9%81%D8%A7%D8%AA%D9%87%D8%A7-%D9%8A%D9%88%D8%B3%D9%81-%D9%85%D9%86%D8%B5%D9%88%D8%B1-%D9%8A%D9%81%D8%AC%D8%B1-%D9%85%D9%81%D8%A7%D8%AC%D8%A3%D8%A9-%D8%B9%D9%86-%D8%B3%D9%88%D8%B2%D8%A7%D9%86-%D8%AA%D9%85%D9%8A%D9%85?amp|title= بعد 15 عامًا على وفاتها.. يوسف منصور يفجر مفاجأة عن سوزان تميم|website=www.masrawy.com}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://cairoscene.com/buzz/stallone-star-in-youssef-mansour-new-movie|title=Sylvester Stallone to Star in Egyptian film with Youssef Mansour|website=CairoScene}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://scoopempire.com/american-film-chosen-guard-shot-egypt-featuring-youssef-mansour/| title=American Film 'The Chosen Guard' to Be Shot in Egypt Featuring Youssef Mansour|first=Scoop|last=Team|date=February 12, 2017}}&lt;/ref&gt;&lt;ref name=&quot;auto2&quot;&gt;{{Cite web| url=https://stepfeed.com/hollywood-film-to-be-shot-in-egypt-with-an-egyptian-cast-0167|title=Hollywood film to be shot in Egypt with an Egyptian cast|first=Rayana|last=Khalaf|date=February 14, 2017| website=StepFeed}}&lt;/ref&gt;&lt;ref&gt;https://al-ain.com/article/yousef-mansour-profile&lt;/ref&gt;&lt;ref&gt;https://m.elwatannews.com/news/details/4113954&lt;/ref&gt;

33 One of the oldest exterior shots in Santa Monica is [[Buster Keaton]]'s ''[[Spite Marriage]]'' (1929) which shows much of 2nd Street. The comedy ''[[It's a Mad, Mad, Mad, Mad World]]'' (1963) included several scenes shot in Santa Monica, including those along the [[California Incline]], which led to the movie's treasure spot, &quot;The Big W&quot;. The [[Sylvester Stallone]] film ''[[Rocky III]]'' (1982) shows [[Rocky Balboa]] and [[Apollo Creed]] training to fight [[Clubber Lang]] by running on the [[Santa Monica State Beach|Santa Monica Beach]], and Stallone's ''[[Demolition Man (film)|Demolition Man]]'' (1993) includes Santa Monica settings. In ''[[Pee-wee's Big Adventure]]'' (1985), the theft of Pee-wee's bike occurs on the Third Street Promenade.&lt;ref name=peewee&gt;{{cite news |last=Martino |first=Alison |date=August 28, 2014 |title=Third Street Promenade Before the Gap Even Existed |url=https://www.lamag.com/citythinkblog/this-is-what-third-street-promenade-looked-like-before-the-gap-even-existed/ |work=CityThink |publisher=Los Angeles Magazine |access-date=February 6, 2019 |archive-date=February 9, 2019 |archive-url=https://web.archive.org/web/20190209124547/https://www.lamag.com/citythinkblog/this-is-what-third-street-promenade-looked-like-before-the-gap-even-existed/ |url-status=live }}&lt;/ref&gt; [[Henry Jaglom]]'s indie ''[[Someone to Love (1987 film)|Someone to Love]]'' (1987), the last film in which [[Orson Welles]] appeared, takes place in Santa Monica's venerable Mayfair Theatre. ''[[Heathers]]'' (1988) used Santa Monica's [[John Adams Middle School, Santa Monica, California|John Adams Middle School]] for many exterior shots. ''[[The Truth About Cats &amp; Dogs]]'' (1996) is set entirely in Santa Monica, particularly the Palisades Park area, and features a radio station that resembles [[KCRW]] at [[Santa Monica College]]. ''[[17 Again (film)|17 Again]]'' (2009) was shot at [[Santa Monica High School|Samohi]]. Other films that show significant exterior shots of Santa Monica include ''[[Fletch (film)|Fletch]]'' (1985), ''[[Species (film)|Species]]'' (1995), ''[[Get Shorty (film)|Get Shorty]]'' (1995), and ''[[Ocean's Eleven (2001 film)|Ocean's Eleven]]'' (2001). [[Richard Rossi]]'s biopic ''Aimee Semple McPherson'' opens and closes at the beach in Santa Monica. ''[[Iron Man (2008 film)|Iron Man]]'' features the Santa Monica pier and surrounding communities as Tony Stark tests his experimental flight suit.

34 During these years, Waits sought to broaden his career beyond music. He befriended actor and director [[Sylvester Stallone]] and made his film debut as a drunken piano player in Stallone's ''[[Paradise Alley]]'' (1978).{{sfnm|1a1=Humphries|1y=2007|1pp=111-112|2a1=Hoskyns|2y=2009|2pp=200-201}} With [[Paul Hampton]], Waits also began writing a movie musical, although this project never came to fruition.{{sfn|Hoskyns|2009|p=204}} Another project he began at this time was a book about entertainers of the past whom he admired.{{sfn|Hoskyns|2009|p=204}}

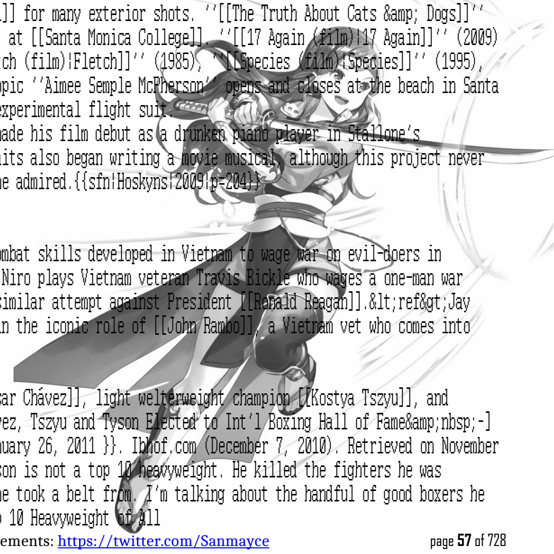
35 * ''[[Over the Top (1987 film)]]'', a 1987 film directed by [[Menahem Golan]], starring [[Sylvester Stallone]]

36 | [[Sylvester Stallone]] || United States

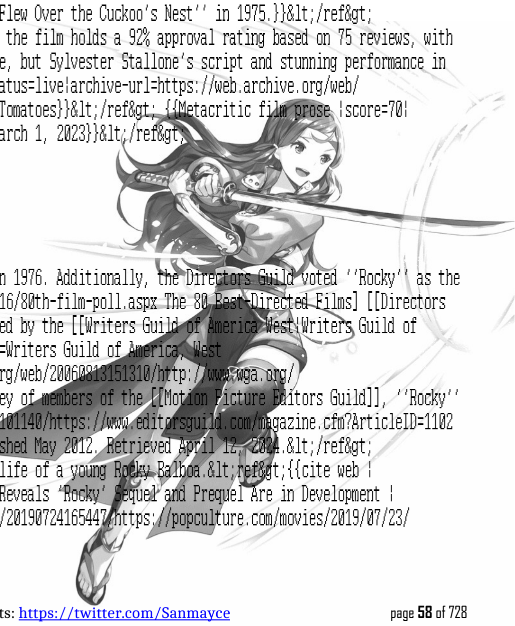
37 In many films, like ''[[Gordon's War]]'' (1973) and ''[[Rolling Thunder (film)|Rolling Thunder]]'' (1977), the veteran uses his combat skills developed in Vietnam to wage war on evil-doers in America.&lt;ref name=&quot;Michael Parris 1987&quot;/&gt; This is also the theme of ''[[Taxi Driver]]'' (1976) in which Robert De Niro plays Vietnam veteran Travis Bickle who wages a one-man war against society whilst he makes plans to assassinate a presidential candidate. This film inspired [[John W. Hinckley]] to make a similar attempt against President [[Ronald Reagan]].&lt;ref&gt;Jay Hyams (1984) ''War Movies'': 197&lt;/ref&gt; In a similar vein is ''[[First Blood]]'' (1982), which stars [[Sylvester Stallone]] in the iconic role of [[John Rambo]], a Vietnam vet who comes into conflict with a small-town police department.

38 [[Sylvester Stallone]] voiced Revere in the animated series ''[[Liberty's Kids]].''

39 On June 12, 2011, Tyson was inducted to the [[International Boxing Hall of Fame]] alongside legendary Mexican champion [[Julio César Chávez]], light welterweight champion [[Kostya Tszyu]], and actor/screenwriter [[Sylvester Stallone]].&lt;ref&gt;[http://www.ibhof.com/pages/inductionweekend/2011/11announce.html Boxers Chavez, Tszyu and Tyson Elected to Int'l Boxing Hall of Fame&amp;nbsp;-] {{webarchive|url=https://web.archive.org/web/20110126212008/http://ibhof.com/pages/inductionweekend/2011/11announce.html |date=January 26, 2011 }}. Ibhof.com (December 7, 2010). Retrieved on November 25, 2011.&lt;/ref&gt; In 2011, [[Bleacher Report]] omitted Tyson from its list of top 10 heavyweights, saying that &quot;Mike Tyson is not a top 10 heavyweight. He killed the fighters he was supposed to beat, but when he fought another elite fighter, he always lost. I'm not talking about some of those B-level fighters he took a belt from. I'm talking about the handful of good boxers he fought throughout his career.&quot;&lt;ref&gt;{{cite web|last=Reeves|first=John|title=Mike Tyson Is No Ali: Why Tyson Is Not a Top 10 Heavyweight of All



```
Time|url=https://bleacherreport.com/articles/582276-mike-tyson-is-no-ali-why-tyson-is-not-a-top-10-heavyweight-of-all-time|access-date=October 13, 2020|website=Bleacher Report|language=en|archive-date=November 11, 2020|archive-url=https://web.archive.org/web/20201111215647/https://bleacherreport.com/articles/582276-mike-tyson-is-no-ali-why-tyson-is-not-a-top-10-heavyweight-of-all-time|url-status=live}}&lt;/ref&gt;
40 The [[Roosevelt Island Tramway]], one of two commuter cable car systems in North America, takes commuters between [[Roosevelt Island]] and Manhattan in less than five minutes, and has been serving the island since 1978.&lt;ref&gt;Lee, Jennifer 8. [https://www.nytimes.com/2006/04/19/nyregion/19roosevelt.html &quot;Midair Rescue Lifts Passengers From Stranded East River Tram&quot;] {{Webarchive|url=https://web.archive.org/web/20160102013419/http://www.nytimes.com/2006/04/19/nyregion/19roosevelt.html |date=January 2, 2016 }}, ''The New York Times'', April 19, 2006. Accessed February 28, 2008. &quot;The system, which calls itself the only aerial commuter tram in the country, has been featured in movies including ''City Slickers'', starring Billy Crystal; ''Nighthawks'', with Sylvester Stallone; and ''Spider-Man'' in 2002.&quot;&lt;/ref&gt;&lt;ref&gt;[http://www.rioc.com/thetram.htm The Roosevelt Island Tram], [[Roosevelt Island Operating Corporation]]. Accessed April 30, 2007. {{webarchive |url=https://web.archive.org/web/20110928212116/http://www.rioc.com/thetram.htm |date=September 28, 2011 }}&lt;/ref&gt;
41 * [[Sylvester Stallone]]
42 File:Sylvester Stallone 1978 (cropped).jpg| [[Sylvester Stallone]]
43 * [[Sylvester Stallone]]
44 * [[Sylvester Stallone]]
45 Gibson also played two villains: Luther Voz in ''[[Machete Kills]]'' in 2013, opposite [[Danny Trejo]], and Conrad Stonebanks in ''[[The Expendables 3]]'' opposite [[Sylvester Stallone]] in 2014.
46 [[File:Mel &amp; Sly Cannes 2013.jpg|thumb|upright|Gibson with ''Expendables'' co-star [[Sylvester Stallone]] (background) in 2014]]
47 [[File:Sylvester Stallone Cannes 2019.jpg|thumb|100px|[[Sylvester Stallone]]]]
48 ** [[Sylvester Stallone]], American actor, screenwriter and film director (''[[Rocky]]'')
49 | writer         = [[Sylvester Stallone]]
50 * Sylvester Stallone
51 '''''Rocky''''' is a 1976 American [[sports film|sports]] [[drama film]] directed by [[John G. Avildsen]] and written by and starring [[Sylvester Stallone]]. It is the first installment in the [[Rocky (film series)|''Rocky'' franchise]] and also stars [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], and [[Burgess Meredith]]. In the film, [[Rocky Balboa]] (Stallone), a poor small-time [[club fighter]] and [[loanshark]] debt collector from [[Philadelphia]], gets an unlikely shot at the world heavyweight championship held by [[Apollo Creed]] (Weathers).
52 [[File:Sylvester_Stallone_-_1977.jpg|thumb|[[Sylvester Stallone]] in 1977]]
53 * [[Sylvester Stallone]] as [[Rocky Balboa|Robert &quot;Rocky&quot; Balboa]]
54 [[Sylvester Stallone]] wrote the screenplay for ''Rocky'' in three and a half days, shortly after watching the [[Muhammad Ali vs. Chuck Wepner|championship match]] between [[Muhammad Ali]] and [[Chuck Wepner]] that took place at [[Richfield Coliseum]] in [[Richfield, Ohio]], on March 24, 1975. Wepner was [[Technical knockout|TKO'd]] in the 15th&amp;nbsp;round of the match by Ali, with few expecting him to last as long as he did. Despite the match motivating Stallone to begin work on ''Rocky'',&lt;ref&gt;{{cite news|title='Rocky Isn't Based on Me,' Says Stallone, 'But We Both Went the Distance'|url=https://www.nytimes.com/packages/html/movies/bestpictures/rocky-ar.html|access-date=December 1, 2015|work=[[The New York Times]]|date=November 1, 1976|archive-url=https://web.archive.org/web/20151201172004/https://www.nytimes.com/packages/html/movies/bestpictures/rocky-ar.html|archive-date=December 1, 2015|url-status=live}}&lt;/ref&gt; he has denied Wepner provided any inspiration for the script.&lt;ref name=&quot;ESPN Wepner Recognized&quot;&gt;{{cite web |url=https://www.espn.com/boxing/story/_/page/IamChuckWepner/chuck-wepner-recognized-rocky-fame |title=Chuck Wepner finally recognized for 'Rocky' fame |publisher=[[ESPN]] |date=October 25, 2011 |access-date=November 20, 2012 |archive-date=September 13, 2014 |archive-url=https://web.archive.org/web/20140913001228/http://espn.go.com/boxing/story/_/page/IamChuckWepner/chuck-wepner-recognized-rocky-fame |url-status=live }}&lt;/ref&gt;&lt;ref name=&quot;The Real Rocky&quot;&gt;{{cite video|people=Feuerzeig, Jeff (Director)|date=October 25, 2011|title=The Real Rocky|medium=Motion picture|publisher=ESPN Films}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Ward |first=Tom |title=The Amazing Story Of The Making Of 'Rocky' |url=https://www.forbes.com/sites/tomward/2017/08/29/the-amazing-story-of-the-making-of-rocky/ |access-date=2023-05-27 |website=Forbes |language=en}}&lt;/ref&gt; Other inspiration for the film may have included characteristics of real-life boxers [[Rocky Marciano]] and [[Joe Frazier]],&lt;ref&gt;{{cite web|last1=Struby|first1=Tim|title=Marciano's career mark unique but flawed?|url=https://www.espn.com/sports/boxing/news/story?id=2165594|publisher=[[ESPN]]|access-date=December 1, 2015|archive-url=https://web.archive.org/web/20151201171638/http://espn.go.com/sports/boxing/news/story?id=2165594|archive-date=December 1, 2015|url-status=live|date=September 21, 2005}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last1=McRae|first1=Donald|title=Still smokin' over Ali but there's no time for hatred now|url=https://www.theguardian.com/sport/2008/nov/11/sportinterviews-boxing|website=[[The Guardian]]|access-date=December 1, 2015|archive-url=https://web.archive.org/web/20151201171407/http://www.theguardian.com/sport/2008/nov/11/sportinterviews-boxing|archive-date=December 1, 2015|date=November 10, 2008}}&lt;/ref&gt; as well as [[Rocky Graziano]]'s autobiography ''Somebody Up There Likes Me'' and [[Somebody Up There Likes Me (1956 film)|the movie of the same name]]. Wepner sued Stallone, and eventually settled for an undisclosed amount.&lt;ref name=&quot;The Real Rocky&quot; /&gt;
55 At the time, Film Artists Management Enterprises (FAME), a joint venture between Hollywood talent agents Craig T. Rumar and Larry Kubik, represented Stallone. He submitted his script to Rumar and Kubik, who immediately saw the potential for it to be made into a motion picture. They shopped the script to various producers and studios in Hollywood but were repeatedly rejected because Stallone insisted that he be cast in the lead role. Eventually, they secured a meeting with Winkler-Chartoff productions (no relation to Henry Winkler). After repeated negotiations with Rumar and Kubik, Winkler-Chartoff agreed to a contract for Stallone to be the writer and also star in the lead role for ''Rocky''.&lt;ref&gt;{{Cite web |date=January 6, 2020 |author=Phil Jay |url=https://www.worldboxingnews.net/2020/06/01/sylvester-stallone-rocky-negotiations/ |title=Exclusive: Sylvester Stallone negotiations for Rocky movie uncovered |website=World Boxing News |access-date=June 10, 2020 |archive-date=June 10, 2020 |archive-url=https://web.archive.org/web/20200610050316/https://www.worldboxingnews.net/2020/06/01/sylvester-stallone-rocky-negotiations/ |url-status=live }}&lt;/ref&gt;
56 [[United Artists]] liked Stallone's script and viewed it as a vehicle for a well-established star like [[Robert Redford]], [[Ryan O'Neal]], [[Burt Reynolds]], or [[James Caan]].&lt;ref&gt;{{cite web|last1=Weisman|first1=Aly|title=Dirt-Poor Sylvester Stallone Turned Down $300,000 In 1976 To Ensure He Could Play 'Rocky'|url=http://www.businessinsider.com/sylvester-stallone-made-rocky-against-all-odds-2014-4|website=[[Business Insider]]|publisher=[[Axel Springer SE]]|access-date=December 1, 2015|date=April 2, 2014|archive-date=December 1, 2015|archive-url=https://web.archive.org/web/20151201172339/http://www.businessinsider.com/sylvester-stallone-made-rocky-against-all-odds-2014-4|url-status=live}}&lt;/ref&gt; Stallone's agents insisted that Stallone portray the title character, to the point of issuing an ultimatum. Stallone later said that he would never have forgiven himself had the film become a success with somebody else in the lead.&lt;ref&gt;{{cite web|url=https://www.forbes.com/sites/tomward/2017/08/29/the-amazing-story-of-the-making-of-rocky/?sh=6bc46fb2560b|title=The Amazing Story Of The Making Of 'Rocky'|first=Tom|last=Ward|date=August 29, 2017|access-date=October 31, 2022|work=[[Forbes]]}}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://archive.nytimes.com/www.nytimes.com/packages/html/movies/bestpictures/rocky-ar.html | title=The New York Times: Best Pictures }}&lt;/ref&gt; He also knew that producers [[Irwin Winkler]] and [[Robert Chartoff]]'s contract with the studio enabled them to &quot;greenlight&quot; a project if the budget was kept low enough. The producers also collateralized any possible losses with their big-budget entry, ''[[New York, New York (1977 film)|New York, New York]]'' (whose eventual losses were covered by ''Rocky''{{'s}} success).&lt;ref name=&quot;Nashawaty&quot;&gt;{{cite magazine |title=EW: The Right Hook: How Rocky Nabbed Best Picture |last=Nashawaty |first=Chris |date=February 19, 2002 |magazine=[[Entertainment Weekly]] |pages=3 |url=https://ew.com/article/2002/02/19/how-rocky-nabbed-best-picture/ |url-status=live |archive-date=November 5, 2014 |archive-url=https://web.archive.org/web/20141105222608/http://www.ew.com/ew/article/0,,203553,00.html |access-date=May 4, 2020 }}&lt;/ref&gt;&lt;ref&gt;[[Neal Gabler]], ''ReelThirteen'', from [http://thirteen.org/ WNET] {{Webarchive|url=https://web.archive.org/web/20140617155123/http://www.thirteen.org/ |date=June 17, 2014 }}, February 22, 2014.&lt;/ref&gt; The film's production budget ended up being $1,075,000, with a further $100,000 spent on producers' fees and $4.2&amp;nbsp;million on advertising costs.&lt;ref name=&quot;BlockWilson583&quot;&gt;{{cite book |editor1-last=Block|editor1-first=Alex Ben|editor2-last=Wilson|editor2-first=Lucy Autrey|year=2010|title=George Lucas's Blockbusting: A Decade-By-Decade Survey of Timeless Movies Including Untold Secrets of Their Financial and Cultural Success|publisher=[[HarperCollins]]|isbn=978-0-06-177889-6|page=[https://books.google.com/books?id=vpbuSXSSqdkC&amp;pg=PA583 583]|quote=The budget was $1,075,000 plus producer's fees of $100,000 ... The advertising costs were $4.2 million, slightly higher than the $4 million UA spent on ads for ''One Flew Over the Cuckoo's Nest'' in 1975.}}&lt;/ref&gt;
57 The film enjoys a reputation as a classic and still receives nearly universal praise. On the [[review aggregator]] [[Rotten Tomatoes]], the film holds a 92% approval rating based on 75 reviews, with an average rating of 8.4/10. The site's critics consensus states: &quot;This story of a down-on-his-luck boxer is thoroughly predictable, but Sylvester Stallone's script and stunning performance in the title role brush aside complaints.&quot;&lt;ref&gt;{{cite web|title=Rocky|url=http://www.rottentomatoes.com/m/1017776-rocky/|url-status=live|archive-url=https://web.archive.org/web/20210228201751/https://www.rottentomatoes.com/m/1017776-rocky|archive-date=February 28, 2021|access-date=April 10, 2022|website=Rotten Tomatoes}}&lt;/ref&gt; {{Metacritic film prose |score=70|count=14}}&lt;ref&gt;{{cite web |title= Rocky Reviews |url= https://www.metacritic.com/movie/rocky |work= [[Metacritic]] |access-date=March 1, 2023}}&lt;/ref&gt;
58 | [[Sylvester Stallone]]
59 | Sylvester Stallone
60 | rowspan=&quot;2&quot;| Sylvester Stallone
61 | Sylvester Stallone
62 | Sylvester Stallone
63 The [[Directors Guild of America]] awarded ''Rocky'' its [[Directors Guild of America Awards|annual award]] for best film of the year in 1976. Additionally, the Directors Guild voted ''Rocky'' as the 65th best-directed film of all time,&lt;ref name=&quot;dga 80 best&quot;&gt;[https://www.dga.org/craft/dgaq/all-articles/1602-spring-2016/80th-film-poll.aspx The 80 Best-Directed Films] [[Directors Guild of America]]. Retrieved April 12, 2024.&lt;/ref&gt; and in 2006, Sylvester Stallone's original screenplay for the film was selected by the [[Writers Guild of America West|Writers Guild of America]] as the 78th best screenplay of all time.&lt;ref name=&quot;wga 101&quot;&gt;{{cite web|title=The 101 Best Screenplays |author=Writers Guild of America, West |url=http://www.wga.org/subpage_newsevents.aspx?id=1807 |access-date=August 24, 2006|url-status=dead |archive-url=https://web.archive.org/web/20060813151310/http://www.wga.org/subpage_newsevents.aspx?id=1807 |archive-date=August 13, 2006 |author-link=Writers Guild of America, West }}&lt;/ref&gt; In a 2012 survey of members of the [[Motion Picture Editors Guild]], ''Rocky'' was voted as one of the 75 best-edited films in all of cinema.&lt;ref name=&quot;mpeg 75&quot;&gt;[https://web.archive.org/web/20150317101140/https://www.editorsguild.com/magazine.cfm?ArticleID=1102 The 75 Best Edited Films] ''[[Motion Picture Editors Guild|Editors Guild Magazine]]'' (Vol. 1, Issue 3) via [[Internet Archive]]. Published May 2012. Retrieved April 12, 2024.&lt;/ref&gt;
64 In July 2019, Stallone said in an interview that there have been ongoing discussions about a prequel to the original film based on the life of a young Rocky Balboa.&lt;ref&gt;{{cite web |url=https://popculture.com/movies/2019/07/23/sylvester-stallone-reveals-rocky-sequel-prequel-in-development/ |title=Sylvester Stallone Reveals 'Rocky' Sequel and Prequel Are in Development |work=popculture.com |date=July 16, 2019 |access-date=July 23, 2019 |archive-date=July 24, 2019 |archive-url=https://web.archive.org/web/20190724165447/https://popculture.com/movies/2019/07/23/sylvester-stallone-reveals-rocky-sequel-prequel-in-development/ |url-status=live }}&lt;/ref&gt;
65 * [https://www.espn.com/page2/s/stallone/011207.html The Making of ''Rocky''] by Sylvester Stallone
66 {{Sylvester Stallone}}
```



```
67 [[Category:Films with screenplays by Sylvester Stallone]]
68 In 1993, Stone played a [[femme fatale]] in the erotic thriller ''[[Sliver (film)|Sliver]]'', based on [[Ira Levin]]'s [[Sliver (novel)|eponymous novel]] about the mysterious occurrences in a privately owned New York City high-rise apartment building. The film was heavily panned by critics and earned Stone a [[Golden Raspberry Awards|Golden Raspberry Award]] nomination for [[Golden Raspberry Award for Worst Actress|Worst Actress]] but became a commercial success, grossing US$116.3&amp;nbsp;million at the international box office.&lt;ref&gt;{{cite web|url=http://www.boxofficemojo.com/movies/?id=sliver.htm|title=Sliver (1993)website=Box Office Mojo|access-date=October 11, 2017|archive-url=https://web.archive.org/web/20171011233153/http://www.boxofficemojo.com/movies/?id=sliver.htm|archive-date=October 11, 2017|url-status=live}}&lt;/ref&gt; She also made a cameo appearance in the action film ''[[Last Action Hero]]'' (1993), reuniting with Arnold Schwarzenegger. In 1994, Stone appeared as the wife of an architect opposite [[Richard Gere]] in the drama ''[[Intersection (1994 film)|Intersection]]'', and as a woman who entices a bomb expert she is involved with into destroying the criminal gang that killed her family, alongside [[Sylvester Stallone]], in the action thriller ''[[The Specialist]]''. While ''Intersection'' found limited success, ''The Specialist'' made US$170.3&amp;nbsp;million worldwide.&lt;ref&gt;{{cite web|url=http://www.boxofficemojo.com/movies/?id=specialist.htm|title=The Specialist (1994)website=Box Office Mojo|access-date=October 11, 2017|archive-url=https://web.archive.org/web/20171011034419/http://www.boxofficemojo.com/movies/?id=specialist.htm|archive-date=October 11, 2017|url-status=live}}&lt;/ref&gt; For her work in both films, she won a [[Golden Raspberry Awards|Golden Raspberry Award]] and a [[Stinkers Bad Movie Awards|Stinkers Bad Movie Award]] for Worst Actress, but was nominated for the [[MTV Movie Award for Most Desirable Female]] for ''The Specialist''.
69 Asia were also credited with contributing the [[Giorgio Moroder]] produced track &quot;Gypsy Soul&quot; to the [[Sylvester Stallone]] film soundtrack to ''[[Over the Top (1987 film)|Over the Top]]'' (1987), although Wetton was the only band member involved.
70 * In the film ''[[Over the Top (1987 film)|Over the Top]]'' with Sylvester Stallone, a song called &quot;Gypsy Soul&quot; is credited to Asia, although John Wetton was the only member involved as the song was written and recorded by [[Giorgio Moroder]].
71 The film was often compared to Verhoeven's previous work on ''RoboCop'', with some reviews remarking that ''Total Recall'' lacked the same &quot;impudence and incandescence&quot; or satirization of 1980s action films as the earlier film.&lt;ref name=&quot;ReviewLATimes&quot;/&gt;&lt;ref name=&quot;ReviewWAPOHowe&quot;/&gt;&lt;ref name=&quot;ReviewEW&quot;/&gt; Some said the film was only fun when Verhoeven inserted moments of ''RoboCop''{{'}}s [[Camp (style)|camp style]].&lt;ref name=&quot;ReviewWAPOHowe&quot;/&gt;&lt;ref name=&quot;ReviewEW&quot;/&gt; ''[[The Washington Post]]''{{'}}s review compared it unfavorably with the [[Sylvester Stallone]] action film ''[[Cobra (1986 film)|Cobra]]'' (1986), saying it was disappointing in its overuse of violence and abandonment of cynicism and creativity for machoism and misogyny.&lt;ref name=&quot;ReviewWAPOKempley&quot;/&gt; Several reviews focused on the excessive violence, with [[Vincent Canby]] describing it as part of an influx of action-adventure films featuring numerous deaths, counting seventy-four kills in the film and over two hundred in ''Die Hard 2''.&lt;ref name=&quot;ReviewNYTimes&quot;/&gt;&lt;ref name=&quot;NYTimesCanbyViolence&quot;/&gt; Some were concerned by the dismissive and sometimes comical depiction of the deaths, and the general reliance on violence as a solution to all problems posed.{{efn|Attributed to multiple references:&lt;ref name=&quot;ReviewLATimes&quot;/&gt;&lt;ref name=&quot;ReviewNYTimes&quot;/&gt;&lt;ref name=&quot;ReviewChicagoTribune&quot;/&gt;&lt;ref name=&quot;ReviewWAPOKempley&quot;/&gt;&lt;ref name=&quot;NYTimesCanbyViolence&quot;/&gt;}} Even so, the ''[[Los Angeles Times]]''{{'}}s review said the violence never seemed to be deliberately sadistic or callous.&lt;ref name=&quot;ReviewLATimes&quot;/&gt; Despite this criticism, Bottin's practical effects were roundly praised, particularly the three-breasted prostitute and mutants that provided many of the film's standout visuals, despite their sometimes perverse or macabre nature.{{efn|Attributed to multiple references:&lt;ref name=&quot;ASCHistory&quot;/&gt;&lt;ref name=&quot;ReviewNYTimes&quot;/&gt;&lt;ref name=&quot;ReviewRollingStone&quot;/&gt;&lt;ref name=&quot;ReviewEW&quot;/&gt;}}
72 Lucas initially resisted casting Ford as Han, since Ford had previously worked with Lucas on ''[[American Graffiti]]'', and was therefore not unknown. Instead, the director asked Ford to assist with auditions by reading lines with other actors. However, Lucas was eventually won over by Ford, and cast him as Han over many other actors who auditioned.{{Efn|Other actors who reportedly auditioned for the role of Han Solo include [[James Caan]], [[Chevy Chase]], [[Robert De Niro]], [[Richard Dreyfuss]], [[Steve Martin]], [[Bill Murray]], [[Jack Nicholson]], [[Nick Nolte]], [[Al Pacino]], [[Burt Reynolds]], [[Kurt Russell]], [[Sylvester Stallone]], [[John Travolta]], [[Christopher Walken]], and [[Perry King]], who later played Solo in the [[Star Wars (radio series)|radio series]].&lt;ref name=&quot;Dreams&quot; /&gt;&lt;ref&gt;{{Cite web |last=Pollard |first=Andrew |date=September 29, 2013 |title=Movie News: James Caan Talks Turning Down Roles in Superman and Star Wars |url=https://www.starburstmagazine.com/movie-news-james-caan |access-date=March 3, 2024 |work=Starburst Magazine |archive-date=February 6, 2022 |archive-url=https://web.archive.org/web/20220206123130/https://www.starburstmagazine.com/movie-news-james-caan |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Evans |first=Bradford |date=February 17, 2011 |title=The Lost Roles of Bill Murray |url=http://splitsider.com/2011/02/the-lost-roles-of-bill-murray/ |archive-url=https://web.archive.org/web/20150520115025/http://splitsider.com/2011/02/the-lost-roles-of-bill-murray |archive-date=May 20, 2015 |access-date=May 25, 2015 |website=Splitsider}}&lt;/ref&gt;&lt;ref name=&quot;casting&quot; /&gt;&lt;ref&gt;{{cite magazine |last=Farr |first=John |date=September 19, 2014 |title=Bill Murray and the Roles That Got Away |url=https://huffingtonpost.com/john-farr/bill-murray-and-the-roles_b_5850434.html |url-status=live |archive-url=https://archive.today/20160111082612/http://www.huffingtonpost.com/john-farr/bill-murray-and-the-roles_b_5850434.html |archive-date=January 11, 2016 |access-date=May 25, 2015 |magazine=[[Huffington Post]]}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Denham |first=Jess |date=September 12, 2014 |title=Star Wars: Al Pacino turned down part of Han Solo over confusing script |url=https://www.independent.co.uk/arts-entertainment/films/news/star-wars-al-pacino-turned-down-part-of-han-solo-because-the-script-was-too-confusing-9723652.html |url-access=subscription |url-status=live |archive-url=https://ghostarchive.org/archive/20220524/https://www.independent.co.uk/arts-entertainment/films/news/star-wars-al-pacino-turned-down-part-of-han-solo-because-the-script-was-too-confusing-9723652.html |archive-date=May 24, 2022 |access-date=August 11, 2021 |work=[[The Independent]]}}&lt;/ref&gt;&lt;ref&gt;{{cite web|first=Alison|last=Nastasi|date=August 5, 2010|url=http://www.cinematical.com/2010/08/05/imagine-that-sly-stallone-auditioned-for-han-solo/|title=Imagine That: Sly Stallone Auditioned for Han Solo|website=Cinematical|access-date=August 6, 2012|archive-url=https://web.archive.org/web/20100806182036/http://www.cinematical.com/2010/08/05/imagine-that-sly-stallone-auditioned-for-han-solo/|archive-date=August 6, 2010}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Huver |first=Scott |date=May 25, 2018 |title=The Forgotten Han Solo Recalls Stepping Into Harrison Ford's Shoes |url=https://www.hollywoodreporter.com/movies/movie-news/star-wars-forgotten-han-solo-recalls-failed-audition-radio-drama-1114765/ |access-date=March 4, 2024 |website=The Hollywood Reporter |language=en-US |archive-date=March 4, 2024 |archive-url=https://web.archive.org/web/20240304045812/https://www.hollywoodreporter.com/movies/movie-news/star-wars-forgotten-han-solo-recalls-failed-audition-radio-drama-1114765/ |url-status=live }}&lt;/ref&gt;}}
73 That same year, Banderas portrayed the antagonist in the [[Richard Donner]]-directed action film ''[[Assassins (1995 film)|Assassins]]'', co-starring opposite [[Sylvester Stallone]] and [[Julianne Moore]].&lt;ref&gt;{{cite web|url= https://www.rottentomatoes.com/m/assassins|title= Assassins|website= Rotten Tomatoes|accessdate= May 31, 2024}}&lt;/ref&gt; In 1996, he starred alongside [[Madonna]] in the musical film ''[[Evita (1996 film)|Evita]]'', an adaptation of the stage musical by [[Andrew Lloyd Webber]] and [[Tim Rice]] in which he played the narrator, Che, a role played by [[David Essex]] in the original 1978 West End production. [[Janet Maslin]] of ''[[The New York Times]]'' wrote that &quot;Banderas{{Nbsp}}... does an unexpectedly splendid job as the film's conspiratorial singing narrator.&quot;&lt;ref&gt;{{cite web|url= https://www.nytimes.com/1996/12/25/movies/madonna-chic-pop-star-as-chic-political-star.|title= Madonna, Chic Pop Star, As Chic Political Star|website= [[The New York Times]]|accessdate= May 31, 2024}}&lt;/ref&gt; For his performance, he was nominated for the [[Golden Globe Award for Best Actor in a Motion Picture - Musical or Comedy]].&lt;ref&gt;{{cite web|url= https://www.playbill.com/article/evita-film-wins-3-golden-globe-awards-com-69350|title= Evita Film Wins 3 Golden Globe Awards|website= Playbill|accessdate= May 31, 2024}}&lt;/ref&gt; He also had success with his role as the masked swordsman [[Zorro]] in the 1998 film ''[[The Mask of Zorro]],'' starring [[Anthony Hopkins]] and [[Catherine Zeta-Jones]]. [[Roger Ebert]] praised the onscreen chemistry between the two leads, writing, &quot;The best scenes in the movie are between Banderas and Zeta-Jones, who share chemistry and, it turns out, a sense of justice.&quot;&lt;ref&gt;{{cite web|url= https://www.rogerebert.com/reviews/the-mask-of-zorro-1998|title= The Mask of Zorro|website= Rogerebert.com|accessdate= May 31, 2024}}&lt;/ref&gt; His performance earned him another [[Golden Globe Award for Best Actor in a Motion Picture - Musical or Comedy|Golden Globe Award]] nomination.&lt;ref&gt;{{cite web|url= https://goldenglobes.com/person/antonio-banderas/|title= Antonio Banderas|website= Golden Globe Awards|accessdate= May 31, 2024}}&lt;/ref&gt; In 1999, he starred in the historical action film ''[[The 13th Warrior]]'', a movie about a [[Muslim]] caught up in a war between the [[Norsemen|Northman]] and human-eating beasts.&lt;ref&gt;{{cite web|url= https://www.rottentomatoes.com/m/13th_warrior|title= The 13th Warrior|website= Rotten Tomatoes|accessdate= May 31, 2024}}&lt;/ref&gt;
74 | Denise Richards&lt;ref&gt;{{cite web|url=http://www.mtv.com/news/2435621/sylvester-stallone-and-denise-richards-nominated-for-razzies-equivalent-the-golden-kela-awards/|title=Sylvester Stallone And Denise Richards Nominated For Razzies Equivalent, The Golden Kela Awards|publisher=MTV |date=February 22, 2010|access-date=June 6, 2020 |archive-date=February 24, 2010 |archive-url=https://web.archive.org/web/20100224081228/http://moviesblog.mtv.com/2010/02/22/sylvester-stallone-and-denise-richards-nominated-for-razzies-equivalent-the-golden-kela-awards/ |url-status=dead }}&lt;/ref&gt;
75 * ''[[Oscar (1991 film)|Oscar]]'', a comedy film directed by [[John Landis]] in which Sylvester Stallone's character is implied to have been at the massacre
76 Many episodes featured actors, such as [[Steve Martin]], [[Harvey Korman]], [[Sylvester Stallone]], [[Julie Andrews]], [[Carol Burnett]], [[Liza Minnelli]], [[Christopher Reeve]], [[Raquel Welch]], [[Joel Grey]] and [[Dom DeLuise]]; some had veteran performers like [[Ethel Merman]], [[Bob Hope]], [[Danny Kaye]], [[Don Knotts]], [[Liberace]], [[Peter Ustinov]], [[James Coburn]], [[Lena Horne]], [[Zero Mostel]] and [[Vincent Price]], and others had well-known pop singers, including [[Elton John]], [[Diana Ross]], [[Harry Belafonte]], [[Kenny Rogers]], [[Linda Ronstadt]], [[Alice Cooper]], [[Paul Simon]], [[Helen Reddy]], [[Debbie Harry]] and [[Leo Sayer]]. Sayer's show used his hit song &quot;[[The Show Must Go On (Leo Sayer song)|The Show Must Go On]]&quot; with slightly changed lyrics in the second verse, from &quot;I wish I could tear down the walls of this theatre&quot; to &quot;I wish I could tear down the walls of this Muppet Theatre&quot;. Some guest stars, such as the [[Monty Python]] star [[John Cleese]], co-wrote much of their own episodes.&lt;ref&gt;{{cite news|title=The 10 best Muppet Show guests|url=https://www.theguardian.com/culture/gallery/2012/jan/22/ten-best-muppets-in-pictures|access-date=30 March 2020|work=The Observer|date=22 January 2012}}&lt;/ref&gt; The second-to-last episode, in 1981, had the then-[[James Bond (character)|James Bond]] actor [[Roger Moore]], while the final episode to be taped guest-starred the actor and dancer [[Gene Kelly]]. [[Mark Hamill]] appeared in one episode as both himself and [[Luke Skywalker]], his role in the ''[[Star Wars]]'' film series. Two of Henson's childhood idols, [[Edgar Bergen]] and [[Milton Berle]], also guest-starred during the second season.
77 Dury had small parts in several films, probably the best known of which was [[Peter Greenaway]]'s ''[[The Cook, the Thief, His Wife &amp; Her Lover]]'' (1989), as well as a [[cameo role|cameo appearance]] in [[Roman Polanski]]'s ''[[Pirates (1986 film)|Pirates]]'' (1986). He also appeared in the [[Eduardo Guedes]] film '' Rocinante'' (1986), the German comedy (lead) ''Brennende Betten'' (Burning Beds) (1988), [[Alejandro Jodorowsky]]'s ''[[The Rainbow Thief]]'' (1990), and the [[Sylvester Stallone]] science fiction film ''[[Judge Dredd (film)|Judge Dredd]]'' (1995). His other film appearances included roles in ''Number One'' (1985) starring [[Bob Geldof]], the [[Bob Hoskins]] film ''[[The Raggedy Rawney]]'' (1988), and ''[[Split Second (1992 film)|Split Second]]'' (1992) starring [[Rutger Hauer]] and [[Kim Cattrall]]. He also appeared alongside fellow lyricists [[Bob Dylan]] and [[Tom Waits]], respectively, in the movies ''[[Hearts of Fire]]'' (1987) and ''Bearskin: An Urban Fairytale'' (1990), also by Eduardo Guedes.&lt;ref&gt;{{cite news |first=Paul |last=Willemen |url=https://www.theguardian.com/news/2000/oct/17/guardianobituaries.filmnews1 |title=Obituary: Eduardo Guedes |work=[[The Guardian]] |date=17 October 2000 |access-date=12 June 2018 |archive-date=12 June 2018 |archive-url=https://web.archive.org/web/20180612145118/https://www.theguardian.com/news/2000/oct/17/guardianobituaries.filmnews1 |url-status=live }}&lt;/ref&gt; His later films included the comedy ''[[Different for Girls (film)|Different for Girls]]'' (1996), and ''[[The Crow: City of Angels]]'' (1996), directed by [[Tim Pope]], who had directed a few of Dury's music videos.{{cn|date=April 2023}}
78 On October 14, 1994, ''Pulp Fiction'' went into general release in the United States. As Peter Biskind describes, &quot;It was not platformed, that is, it did not open in a handful of theaters and roll out slowly as word of mouth built, the traditional way of releasing an [[Independent film|indie film]]; it went wide immediately, into 1,100 theaters.&quot;{{sfn|Biskind|2004|p=189}} In the eyes of some cultural critics, ''Reservoir Dogs'' had given Tarantino a reputation for glamorizing violence. Miramax played with the issue in its marketing campaign: &quot;You won't know the facts till
```

you've seen the fiction&quot;, went one slogan.{{sfn¦Dawson¦1995a¦p=171}} ''Pulp Fiction'' was the [[List of 1994 box office number-one films in the United States¦top-grossing film at the US box office]] its first weekend with a gross of $9,311,882, edging out a [[Sylvester Stallone]] vehicle, ''[[The Specialist]]'', which was in its second week and playing at more than twice as many theaters. The gross claimed by Miramax was disputed by others. [[Warner Bros.]] initially reported an estimated gross of $8.9&amp;nbsp;million for ''The Specialist'' with Bob Weinstein then reporting a gross for ''Pulp Fiction'' of $9.1&amp;nbsp;million, claiming that the film was on another 100 screens that had previously been overlooked. Warners then updated their gross to $9.3&amp;nbsp;million, claiming they had made a calculation error.&lt;ref&gt;{{cite magazine¦magazine=[[Daily Variety]]¦page=1¦last=Klady¦first=Leonard¦date=October 17, 1994¦title=Shootout at the B.O. peak}}&lt;/ref&gt;

79 In the 1980s, Emerson began to write and perform music for films, as his orchestral and classical style was more suited for film work than for the [[new wave music¦new wave]]-dominated pop/ rock market.&lt;ref name=perryhonky /&gt; He was given the script for ''[[Chariots of Fire]]'', but turned down the offer to score it.&lt;ref name=EMM83/&gt; Films for which Emerson contributed soundtrack music include [[Dario Argento]]'s ''[[Inferno (1980 film)¦Inferno]]'' (1980), the action thriller ''[[Nighthawks (1981 film)¦Nighthawks]]'' (1981) starring [[Sylvester Stallone]], (1984 film) Best Revenge, notable because he collaborated with [[Brad Delp]] from the rock band [[Boston (band)¦Boston]] and [[Levon Helm]] from [[The Band]] both on vocals, and [[Garth Hudson]] also from The Band on accordion, that also featured an instrumental piece called &quot;Dream Runner&quot; that became a standard solo performance piece for Emerson during at ELP shows throughout the next decade, [[Lucio Fulci]]'s ''[[Murder Rock]]'' (1984), and Michele Soavi's ''[[The Church (1989 film)¦The Church]]'' (also known as ''La chiesa'') (1989).&lt;ref name=martin&gt;{{cite web ¦last=Martin ¦first=Jeffery X ¦date=6 February 2015 ¦title=Music Review: Keith Emerson, &quot;At the Movies&quot; ¦url=http://popshifter.com/2015-02-06/music-review-keith-emerson-movies/ ¦publisher=Popshifter ¦ website=Popshifter.com ¦access-date=12 March 2016 ¦url-status=dead ¦archive-url=https://web.archive.org/web/20160312222504/http://popshifter.com/2015-02-06/music-review-keith-emerson-movies/ ¦ archive-date=12 March 2016 }}&lt;/ref&gt; He was also the composer for the short-lived 1994 US animated television series ''[[Iron Man (TV series)¦Iron Man]]''.&lt;ref name=hammondhof /&gt;&lt;ref&gt;Macan, ''Endless Enigma'', p. 699.&lt;/ref&gt;

80 [[File:John Rambo.jpg¦thumb¦[[Sylvester Stallone]] as [[John Rambo]] in ''[[Rambo III]]'' (1988). The film was released in the decade where the term action became its own unique genre routinely in terms of promotion and film reviews.{{sfn¦Kendrick¦2019¦p=36}}]]

81 Hauer made his American debut in the [[Sylvester Stallone]] film ''[[Nighthawks (1981 film)¦Nighthawks]]'' (1981) as a psychopathic and cold-blooded terrorist named Wulfgar.&lt;ref name=&quot;:0&quot; /&gt; With his sights set on a long-term career in Hollywood, Hauer worked with an [[Dialect coach¦accent coach]] in the early 1980s to develop a convincing American accent.&lt;ref&gt;{{cite news ¦last=Mann ¦first=Roderick ¦date=20 May 1985 ¦title=For Dutch Actor Hauer, the Accent Is Real ¦work=The Los Angeles Times ¦page=V1}}&lt;/ref&gt; Unafraid of controversial roles,&lt;ref name=NOSsteenhoff/&gt; he portrayed [[Albert Speer]] in the 1982 [[American Broadcasting Company]] production ''[[Inside the Third Reich (film)¦Inside the Third Reich]]''.&lt;ref&gt;{{cite web¦url =https://www.latimes.com/obituaries/story/2019-07-24/rutger-hauer-dead-blade-runner¦title =Dutch actor Rutger Hauer, 'Blade Runner' villain, dies at 75¦first =Nardine¦last =Saad¦website =Los Angeles Times¦date =24 July 2019¦access-date =26 July 2019¦archive-date =25 July 2019¦archive-url =https://web.archive.org/web/20190725164809/https://www.latimes.com/obituaries/story/2019-07-24/rutger-hauer-dead-blade-runner¦url-status =live}}&lt;/ref&gt; The same year, Hauer appeared in arguably his most famous and acclaimed role as the eccentric and violent but sympathetic [[antihero]] [[List of Blade Runner characters#Roy Batty¦Roy Batty]] in [[Ridley Scott]]'s 1982 science fiction thriller ''[[Blade Runner]]'', in which he delivered the famous [[tears in rain monologue]].&lt;ref&gt;{{Cite web¦url=http://www.bbc.co.uk/programmes/profiles/1NZrTvN4jK47KzPY7YH6b37/rutger-hauer¦ title=BBC Two - Tomorrow's Worlds: The Unearthly History of Science Fiction - Rutger Hauer¦website=bbc.co.uk¦access-date=17 January 2018¦archive-date=13 July 2018¦archive-url=https://web.archive.org/web/20180713231215/http://www.bbc.co.uk/programmes/profiles/1NZrTvN4jK47KzPY7YH6b37/rutger-hauer¦url-status=live}}&lt;/ref&gt; Hauer composed parts of the monologue the evening prior to filming, &quot;cutting away swathes of the original script before adding the speech's poignant final line&quot;.&lt;ref&gt;{{Cite web¦url=https://www.radiotimes.com/news/film/2019-07-25/blade-runner-tears-in-rain-speech/¦title=Rutger Hauer dissects his iconic &quot;tears in rain&quot; Blade Runner monologue¦website=Radio Times¦ access-date=25 July 2019¦archive-date=20 July 2018¦archive-url=https://web.archive.org/web/20180720081037/https://www.radiotimes.com/news/film/2017-10-05/blade-runner-tears-in-rain-speech/¦url-status=live}}&lt;/ref&gt; He went on to play the adventurer courting [[Theresa Russell]] in ''[[Eureka (1983 film)¦Eureka]]'' (1983),&lt;ref&gt;{{Cite web¦url=https://variety.com/2019/film/news/rutger-hauer-dies-five-essential-performances-blade-runner-1203278586/¦title=Rutger Hauer: Five of the 'Blade Runner' Star's Essential Performances¦ last=Gaydos¦first=Steven¦date=24 July 2019¦website=Variety¦language=en¦access-date=25 July 2019¦archive-date=25 July 2019¦archive-url=https://web.archive.org/web/20190725000229/https://variety.com/2019/film/news/rutger-hauer-dies-five-essential-performances-blade-runner-1203278586/¦url-status=live}}&lt;/ref&gt; investigative reporter opposite [[John Hurt]] in ''[[The Osterman Weekend (film)¦ The Osterman Weekend]]'' (1983),&lt;ref name=&quot;:3&quot;&gt;{{Cite web¦url=https://www.hollywoodreporter.com/news/rutger-hauer-dead-blade-runner-actor-was-75-1226629¦title=Rutger Hauer, 'Blade Runner' Actor, Dies at 75¦last1=Barnes¦first1=Mike¦last2=Parker¦first2=Ryan¦date=24 July 2019¦website=The Hollywood Reporter¦language=en¦access-date=25 July 2019¦archive-date=25 July 2019¦archive-url=https://web.archive.org/web/20190725001602/https://www.hollywoodreporter.com/news/rutger-hauer-dead-blade-runner-actor-was-75-1226629¦url-status=live}}&lt;/ref&gt; hardened mercenary Martin in ''[[Flesh and Blood (1985 film)¦Flesh &amp; Blood]]'' (1985), and knight paired with [[Michelle Pfeiffer]] in ''[[Ladyhawke (film)¦Ladyhawke]]'' (1985).&lt;ref name=&quot;:3&quot; /&gt;

82 [[Stan Bush]]'s song &quot;[[The Touch (Stan Bush song)¦The Touch]]&quot; is prominently featured in the film, having been originally written for the [[Sylvester Stallone]] film ''[[Cobra (1986 film)¦Cobra]]'' (1986).&lt;ref&gt;{{cite web¦url=https://www.songfacts.com/facts/stan-bush/the-touch ¦title=The Touch Songfacts ¦publisher=Songfacts.com ¦access-date=April 27, 2010}}&lt;/ref&gt; A remix is featured in the 2012 video game ''[[Transformers: Fall of Cybertron]]'';&lt;ref&gt;{{cite web¦url=https://www.complex.com/pop-culture/2013/10/the-best-licensed-songs-used-in-video-game-commercials/you-got-the-touch¦title=The Best Licensed Songs Used in Video Game Commercials&quot;You Got the Touch&quot; - Stan Bush¦website=Complex}}&lt;/ref&gt; the song is featured in the 2018 film ''[[Bumblebee (film)¦Bumblebee]]''. The soundtrack includes &quot;Instruments of Destruction&quot; by NRG,&lt;ref&gt;{{cite news¦title= Transformers The Movie - Original Motion Picture Soundtrack 20th Anniversary Edition¦work= IGN¦url= http://uk.music.ign.com/articles/792/792472p1.html¦archive-url= https://web.archive.org/web/20070629003918/http://uk.music.ign.com/articles/792/792472p1.html¦ url-status= dead¦archive-date= June 29, 2007¦access-date=January 11, 2016}}&lt;/ref&gt; &quot;Dare&quot; by Stan Bush,&lt;ref&gt;{{cite web¦url=https://www.space.com/41671-transformers-the-movie-theater-rerelease-date.html¦title='Transformers: The Movie' Re-Release Has the Touch, the Power and 300+ More Theaters¦author=Newsarama Staff ¦website=Space.com¦date=August 30, 2018}}&lt;/ref&gt; &quot;Nothin's Gonna Stand in Our Way&quot; and &quot;Hunger&quot; by [[Kick Axe]] (credited as Spectre General),&lt;ref&gt;{{cite web ¦url=http://www.kickaxe.net/kickaxe_song_view.php ¦title=Kick Axe Songs ¦publisher=Kickaxe.net ¦access-date=May 24, 2017 ¦archive-date=January 20, 2017 ¦archive-url=https://web.archive.org/web/20170120094902/http://www.kickaxe.net/kickaxe_song_view.php ¦url-status=dead }}&lt;/ref&gt; &quot;[[Dare to Be Stupid (song)¦Dare to Be Stupid]]&quot; by [[&quot;Weird Al&quot; Yankovic]], and a hard rock remake of the ''Transformers'' TV theme song by [[Lion (band)¦Lion]].&lt;ref&gt;{{cite web ¦url=http://www.lyricsondemand.com/soundtracks/t/thetransformerslyrics/thetransformersthemelyrics.html ¦title=The Transformers (Theme) Lyrics by Lion - The Transformers Soundtrack Lyrics ¦publisher=Lyricsondemand.com ¦access-date=May 24, 2017 ¦archive-date=September 21, 2024 ¦archive-url=https://web.archive.org/web/20240921022527/https://www.lyricsondemand.com/soundtracks/t/thetransformerslyrics/thetransformersthemelyrics.html ¦url-status=live }}&lt;/ref&gt;

83 Keitel starred in [[Quentin Tarantino]]'s ''[[Reservoir Dogs]]'' (which he co-produced) in 1992,&lt;ref name=tca/&gt; where his performance as &quot;Mr. White&quot; took his career to a different level. Since then, Keitel has chosen his roles with care, seeking to change his image and show a broader acting range.&lt;ref name=tca/&gt; One of those roles was the title character in ''[[Bad Lieutenant]]'', about a self-loathing, drug-addicted police lieutenant trying to redeem himself.&lt;ref name=tca/&gt; He co-starred in the [[Jane Campion]] film ''[[The Piano]]'' in 1993,&lt;ref name=tca/&gt; and played Winston &quot;The Wolf&quot; Wolf in Quentin Tarantino's ''[[Pulp Fiction]]'', an apparent reprise of his Victor the Cleaner character from 1993's [[Point of No Return (1993 film)¦''Point of No Return'']]. Keitel starred as a police detective in [[Spike Lee]]'s ''[[Clockers (film)¦Clockers]]'' (an adaptation of [[Richard Price]]'s novel, co-produced by Martin Scorsese). In 1996, Keitel had a major role in Quentin Tarantino and [[Robert Rodriguez]]'s film ''[[From Dusk till Dawn]]'', and in 1997, he starred in the crime drama ''[[Cop Land]]'', which also starred [[Sylvester Stallone]], [[Ray Liotta]] and Robert De Niro.

84 He became a popular action film star and followed up with commercially successful films such as ''[[Cyborg (film)¦Cyborg]]'', [[Kickboxer (film)¦''Kickboxer'']] (both 1989), ''[[Lionheart (1990 film)¦Lionheart]]'', ''[[Death Warrant (film)¦Death Warrant]]'' (both 1990), ''[[Double Impact]]'' (1991), ''[[Universal Soldier (1992 film)¦Universal Soldier]]'' (1992), ''[[Nowhere to Run (1993 film)¦Nowhere to Run]]'', ''[[Hard Target]]'' (both 1993), ''[[Timecop]]'', ''[[Street Fighter (1994 film)¦Street Fighter]]'' (both 1994), [[Sudden Death (1995 film)¦''Sudden Death'']] (1995), [[The Quest (1996 film)¦''The Quest'']], which marked his [[directorial debut]], and ''[[Maximum Risk]]'' (both 1996). Between 1993 and 1998, three Hong Kong filmmakers collaborated with Van Damme for their debuts in Western cinema: [[John Woo]], [[Tsui Hark]], and [[Ringo Lam]].&lt;ref&gt;{{cite web ¦ url=https://thedissolve.com/features/movie-of-the-week/1088-when-jean-claude-van-damme-became-hong-kongs-gatew/ ¦ title=When Jean-Claude van Damme became Hong Kong's gateway to Hollywood }}&lt;/ref&gt; After a decline in popularity in the late 1990s and early 2000s, he returned to prominence with the critically acclaimed crime drama ''[[JCVD (film)¦JCVD]]'' (2008). His big return to the action genre was in ''[[The Expendables 2]]'' (2012), in which he starred as the villain, opposite [[Sylvester Stallone]]. From thereon, he continued starring in action films and doing extensive voice work.

85 Among his acting heroes growing up were [[Charles Bronson]], [[Bruce Lee]], [[James Dean]], [[Montgomery Clift]], [[Steve McQueen]], [[Jean-Paul Belmondo]], and [[Sylvester Stallone]].&lt;ref&gt;{{Cite web¦url=https://www.undertheradarmag.com/interviews/jean-claude_van_damme_and_the_creators_of_jean-claude_van_johnson¦title=Jean-Claude Van Damme and the Creators of &quot;Jean-Claude Van Johnson&quot; &amp;#124; Under the Radar &amp;#124; Music Blog for the Indie Music Magazine}}&lt;/ref&gt;&lt;ref&gt;{{cite news ¦ url=https://www.theguardian.com/film/2012/aug/10/van-damme-expendables-2 ¦ title=Jean-Claude van Damme: 'I tried to play the system; I was blacklisted' ¦ newspaper=The Guardian ¦ date=10 August 2012 ¦ last1=Godfrey ¦ first1=Alex }}&lt;/ref&gt;

86 Van Damme achieved [[sex symbol]] status through publicity venues in the late 1980's and early 1990's. Unlike his major rivals in action film, such as [[Arnold Schwarzenegger]], [[Sylvester Stallone]], [[Steven Seagal]], and [[Bruce Willis]], Van Damme was nominated by the ''[[National Enquirer]]'' as number one on their &quot;Top Ten Sexiest Men in the World&quot; list, being the only action film star listed.&lt;ref&gt;{{cite book ¦ url=https://books.google.com/books?id=ttxW9ZoE7mwC&amp;dq=Van+Damme+produced+himself+as+an+erotically+charged+persona+and+achieved+sex+symbol+status+through+publicity+venues+in+the+late+1980s+and+early+1990s&amp;pg=PA39 ¦ title=Detecting Men: Masculinity and the Hollywood Detective Film ¦ isbn=978-0-7914-8138-7 ¦ last1=Gates ¦ first1=Philippa ¦ date=February 2012 ¦ publisher=State University of New York Press }}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://www.huffpost.com/entry/jean-claude-van-damme-get_b_141745¦title=Jean-Claude Van Damme Gets &quot;Raw&quot; in JCVD¦date=December 7, 2008¦ website=HuffPost}}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://brightside.me/articles/former-sex-symbol-jean-claude-van-damme-has-posted-a-topless-pic-of-him-at-62-that-is-nothing-but-hot-815507/¦ title=Former Sex Symbol Jean-Claude Van Damme Has Posted a Topless Pic of Him at 62 That Is Nothing but Hot¦date=September 28, 2023¦website=Bright Side - Inspiration. Creativity. Wonder.}}&lt;/ref&gt;

87 [[File:Sylvester Stallone &amp; Bruce Willis (4840665852).jpg¦thumb¦Willis in 2010 with ''[[The Expendables (franchise)¦The Expendables]]'' co-star [[Sylvester Stallone]]]]

88 As of 2010, Willis was the eighth highest-grossing actor in a leading role and 12th-highest including supporting roles.&lt;ref name=&quot;BOMlist&quot;&gt;{{cite web ¦title=People Index ¦ url=https://www.boxofficemojo.com/people/?view=Actor&amp;sort=sumgross&amp;p=.htm ¦url-status=dead ¦archive-url=https://web.archive.org/web/20190627001804/https://www.boxofficemojo.com/people/?view=Actor&amp;sort=sumgross&amp;p=.htm ¦archive-date=June 27, 2019 ¦access-date=August 29, 2010 ¦website=[[Box Office Mojo]]}}&lt;/ref&gt;&lt;ref name=&quot;The Numbers&quot;&gt;{{cite web ¦ title=All Time Top 100 Stars at the Box Office ¦url=https://www.the-numbers.com/people/records/ ¦url-status=dead ¦archive-url=https://web.archive.org/web/20100825163052/http://www.the-numbers.com/people/records/ ¦archive-date=August 25, 2010 ¦access-date=August 29, 2010 ¦publisher=The Numbers}}&lt;/ref&gt; Willis starred with [[Tracy Morgan]] in the 2010 comedy ''[[Cop Out (2010 film)¦Cop Out]]'', directed by [[Kevin Smith]], about two police detectives investigating the theft of a baseball card.&lt;ref name=&quot;CoupleDicks&quot;&gt;{{cite web ¦title=Bruce Willis Circling Several

New Movies |work=[[Empire (magazine)|Empire]] |url=https://www.empireonline.com/news/story.asp?NID=24691 |access-date=May 9, 2009 |archive-date=October 17, 2012 |archive-url=https://web.archive.org/web/20121017124201/http://www.empireonline.com/news/story.asp?NID=24691 |url-status=live }}&lt;/ref&gt; Willis appeared in the music video for the song &quot;[[Stylo (song)|Stylo]]&quot; by [[Gorillaz]].&lt;ref name=&quot;BruceWillisStylo&quot;&gt;{{cite magazine|title=Bruce Willis takes aim at Gorillaz in Stylo video|magazine=[[Billboard (magazine)|Billboard]] |url=https://www.billboard.com/articles/columns/viral-videos/959192/bruce-willis-takes-aim-at-gorillaz-in-stylo-video|access-date=March 2, 2009}}&lt;/ref&gt; Also in 2010, he appeared in a cameo with the former [[Planet Hollywood]] co-owners and 80s action stars [[Sylvester Stallone]] and [[Arnold Schwarzenegger]] in the film ''[[The Expendables (2010 film)|The Expendables]]''. Willis played the role of CIA agent &quot;[[List of The Expendables characters#Mr. Church|Mr. Church]]&quot;. It was the first time the three action stars had appeared on screen together. Although the scene featuring the three was short, it was one of the most highly anticipated scenes in the film. The trio filmed their scene in an empty church on October 24, 2009.&lt;ref&gt;{{Cite web | url=http://www.newsinfilm.com/2009/10/25/stallone-on-rocky-7-plus-expendables-update/ |title=Stallone Shot a Scene with Arnold and Bruce |archiveurl=https://web.archive.org/web/20091027222605/http://www.newsinfilm.com/2009/10/25/stallone-on-rocky-7-plus-expendables-update/ |archivedate=October 27, 2009 |website=News in Film}}&lt;/ref&gt; Willis next starred in ''[[RED (film)|RED]]'', an adaptation of the comic book mini-series [[Red (WildStorm comics)|of the same name]], in which he portrayed Frank Moses. The film was released on October 15, 2010.&lt;ref name=&quot;red&quot;&gt;{{cite web|url=https://www.slashfilm.com/507193/red-begins-principal-photography/|title=Red Begins Principal Photography|publisher=[[/Film]]|date=January 18, 2010|access-date=March 30, 2022|archive-url=https://archive.today/20120914060230/http://www.slashfilm.com/red-begins-principal-photography/|archive-date=September 14, 2012|url-status=live}}&lt;/ref&gt;

89 Willis starred alongside [[Bill Murray]], [[Edward Norton]], and [[Frances McDormand]] in ''[[Moonrise Kingdom]]'' (2012). Filming took place in [[Rhode Island]] under the direction of [[Wes Anderson]], in 2011.&lt;ref&gt;{{cite news|title=Film starring Bruce Willis to be shot in RI|url=http://archive.boston.com/news/local/rhode_island/articles/2011/03/24/film_starring_bruce_willis_to_be_shot_in_ri/|access-date=April 6, 2011|newspaper=[[The Boston Globe]]|date=March 24, 2011|agency=[[Associated Press]]|location=Providence, R.I.|quote=Steven Feinberg, executive director for the Rhode Island Film and Television Office, says on-site work on the film, 'Moonrise Kingdom,' is scheduled to begin this spring. Feinberg says the film will be shot in several locations in Rhode Island.|archive-date=June 25, 2018|archive-url=https://web.archive.org/web/20180625201016/http://archive.boston.com/news/local/rhode_island/articles/2011/03/24/film_starring_bruce_willis_to_be_shot_in_ri/|url-status=live}}&lt;/ref&gt; Willis returned, in an expanded role, in ''[[The Expendables 2]]'' (2012).&lt;ref&gt;{{cite web|date=August 29, 2010 | author=Jason Barr |title=Sylvester Stallone Wants Bruce Willis to Play a &quot;Super Villain&quot; in THE EXPENDABLES Sequel |website=[[Collider (website)|Collider]] |url=https://www.collider.com/2010/08/29/the-expendables-sequel-sylvester-stallone-bruce-willis-super-villain/ |url-status=dead |archive-url=https://web.archive.org/web/20120212053759/http://collider.com/the-expendables-sequel-sylvester-stallone-bruce-willis-super-villain/46297/ |archive-date=February 12, 2012 }}&lt;/ref&gt; He appeared alongside [[Joseph Gordon-Levitt]] in the sci-fi action film ''[[Looper (film)|Looper]]'' (2012), as the older version of Gordon-Levitt's character, Joe.

90 Willis owns houses in Los Angeles and [[Penns Grove, New Jersey]]. He also rents apartments at [[Trump Tower]]&lt;ref&gt;{{Cite news|first=Marc|last=Ambinder|date=November 18, 2016|url=https://www.washingtonpost.com/posteverything/wp/2016/11/18/how-donald-trump-will-retrofit-midtown-manhattan-as-a-presidential-getaway/|title=How Donald Trump will retrofit Midtown Manhattan as a presidential getaway|newspaper=[[The Washington Post]]|access-date=November 2, 2018|archive-date=October 30, 2019|archive-url=https://web.archive.org/web/20191030063252/https://www.washingtonpost.com/posteverything/wp/2016/11/18/how-donald-trump-will-retrofit-midtown-manhattan-as-a-presidential-getaway/|url-status=live}}&lt;/ref&gt; and in [[Riverside South, Manhattan]].&lt;ref name=&quot;TrumpPlace&quot;&gt;{{cite news|last=Abelson |first=Max |title=Bruce Willis Pays $4.26 M. for Trump Enemy's Condo |work=[[The New York Observer]] |url=http://www.observer.com/2007/bruce-willis-pays-4-26-m-trump-enemys-condo |date=November 5, 2007 |access-date=May 10, 2009 |url-status=dead |archive-url=https://web.archive.org/web/20090211070709/http://www.observer.com/2007/bruce-willis-pays-4-26-m-trump-enemys-condo |archive-date=February 11, 2009 }}&lt;/ref&gt; In 2000, Willis and his business partner [[Arnold Rifkin]] started a motion picture production company called [[Cheyenne Enterprises]]. He left the company to be run solely by Rifkin in 2007 after ''Live Free or Die Hard''.&lt;ref name=&quot;Holly2&quot;&gt;{{cite news|last=Fleming |first=Michael |title=Willis held 'Hostage' |work=Variety |url=http://www.accessmylibrary.com/coms2/summary_0286-26789071_ITM |url-access=registration |date=November 12, 2002 |access-date=May 10, 2009 }}{{dead link|date=June 2016|bot=medic}}{{cbignore|bot=medic}}&lt;/ref&gt; He also owns several small businesses in [[Hailey, Idaho]], including The Mint Bar and The Liberty Theater and was one of the first promoters of [[Planet Hollywood]], with actors [[Arnold Schwarzenegger]] and [[Sylvester Stallone]].&lt;ref name=&quot;PlanetH&quot;&gt;{{cite news |last=Martinson |first=Jane |author2=Vikram Dodd |title=Planet Hollywood crashes to earth |work=The Guardian |location=London |url=https://www.theguardian.com/uk/1999/aug/18/janemartinson.vikramdodd |date=August 18, 1999 |access-date=May 10, 2009 |archive-date=November 5, 2013 |archive-url=https://web.archive.org/web/20131105061633/http://www.theguardian.com/uk/1999/aug/18/janemartinson.vikramdodd |url-status=live }}&lt;/ref&gt; Willis and the other actors were paid for their appearances and endorsements through an [[employee stock ownership plan]].&lt;ref&gt;{{Cite news |url=https://economictimes.indiatimes.com/stars-like-bruce-willis-arnold-schwarzenegger-and-sylvester-stallone-endorse-planet-hollywood/articleshow/9937771.cms |title=Stars like Bruce Willis, Arnold Schwarzenegger and Sylvester Stallone endorse Planet Hollywood |date=September 11, 2011 | work=The Economic Times |access-date=June 22, 2018 |publisher=Bennett, Coleman &amp; Co. Ltd. |archive-date=June 23, 2018 |archive-url=https://web.archive.org/web/20180623004904/https://economictimes.indiatimes.com/stars-like-bruce-willis-arnold-schwarzenegger-and-sylvester-stallone-endorse-planet-hollywood/articleshow/9937771.cms |url-status=live }}&lt;/ref&gt;

91 | publisher=Max Boxing| date=2007-02-28| access-date=2007-09-18 |archive-url = https://web.archive.org/web/20071109090947/http://www.maxboxing.com/groves/groves022807.asp &lt;!-- Bot retrieved archive --&gt; |archive-date = 2007-11-09}}&lt;/ref&gt; Also included in that streak was future world champion [[Leo Cruz]], beaten in 13 rounds at San Juan and Derrik Holmes, knocked out in five rounds in a fight attended, among others, by [[Sylvester Stallone]], [[Carl Weathers]] and Alexis Arguello.&lt;ref name=&quot;youtube.com&quot;&gt;[https://www.youtube.com/watch?v=g5A1lj2wCcQ YouTube]. YouTube. Retrieved on 2017-07-14.&lt;/ref&gt; After recording his 32nd. knockout win in a row, he moved up in weight to face the world featherweight champion [[Salvador Sánchez]] of Mexico. [[Salvador Sánchez vs. Wilfredo Gómez|He lost to Salvador Sanchez by 8th-round TKO]].

92 His greatest success came with ''[[Rocky]]'' (1976), which he directed working in conjunction with writer and star [[Sylvester Stallone]].&lt;ref&gt;{{cite book |last1=Powell |first1=Larry | last2=Garrett |first2=Tom |date=2013 |title=The Films of John G. Avildsen: Rocky, The Karate Kid and Other Underdogs |url=https://books.google.com/books?id=8aNTAgAAQBAJ&amp;pg=PA94 | location=Jefferson, North Carolina |publisher=[[McFarland &amp; Company]] |page=94 |isbn=978-0-7864-6692-4 |access-date=2021-06-24 |archive-date=2021-06-24 |archive-url=https://web.archive.org/web/20210624214318/https://books.google.com/books?id=8aNTAgAAQBAJ&amp;pg=PA94 |url-status=live }}&lt;/ref&gt; The film was a major critical and commercial success, becoming the [[1976 in film|highest-grossing film of 1976]] and garnering ten Academy Award nominations and winning three, including Best Picture and Best Director.{{sfn|Powell|Garrett|2013|pp=83-84}} Avildsen later returned to direct what was then expected to be the series' final installment, ''[[Rocky V]]'' (1990).{{sfn|Powell|Garrett|2013|pp=186-192}}

93 A documentary on the life, career and films of Avildsen was released in August 2017, approximately two months after his death. ''John G. Avildsen: King of the Underdogs'' (2017), directed and produced by [[Derek Wayne Johnson]],&lt;ref&gt;{{cite news |last=Drown |first=Michelle |date=January 26, 2017 |title=John Avildsen: King of the Underdogs |url=https://www.independent.com/2017/01/26/john-avildsen-king-underdogs/ |work=[[Santa Barbara Independent]] |access-date=June 16, 2017 |archive-date=June 26, 2019 |archive-url=https://web.archive.org/web/20190626194010/https://www.independent.com/2017/01/26/john-avildsen-king-underdogs/ |url-status=live }}&lt;/ref&gt; features interviews with [[Sylvester Stallone]], [[Ralph Macchio]], [[Martin Scorsese]], [[Jerry Weintraub]], and [[Burt Reynolds]], among others.&lt;ref&gt;{{cite news |last=Farber |first=Stephen |date=February 8, 2017 |title='John G. Avildsen: King of the Underdogs': Film Review {{!}} Santa Barbara 2017 |url=https://www.hollywoodreporter.com/news/general-news/john-g-avildsen-king-underdogs-review-973174/ |work=[[The Hollywood Reporter]] |access-date=June 16, 2017 |archive-date=September 2, 2023 |archive-url=https://web.archive.org/web/20230902050048/https://www.hollywoodreporter.com/movies/movie-reviews/john-g-avildsen-king-underdogs-review-973174/ |url-status=live }}&lt;/ref&gt; The documentary is a companion to the book ''The Films of John G. Avildsen: Rocky, The Karate Kid, and other Underdogs'', written by Larry Powell and Tom Garrett.{{sfn|Powell|Garrett|2013|p=}}

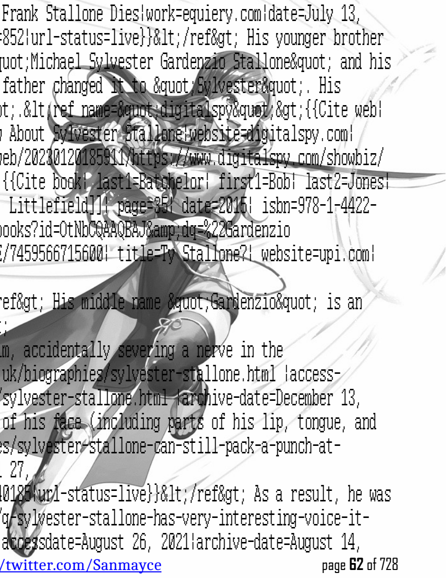
94 * ''Pt'it fromage qui pue.'' (Little stinking cheese) used by M. Sylvestre ([[Sylvester Stallone]]'s puppet) when talking of French people or when saluting PPD.

95 *'''Sylvestre''', '''Commandant Sylvestre''', '''Cardinal Sylvestre, and many others,''' both named and unnamed, all with the same face and voice, were fictional characters based on the likeness of the American actor [[Sylvester Stallone]], although when it was the actor himself who was represented, or [[John Rambo|Rambo]], he had a distinct appearance and a different voice. The Sylvestres were parodies of &quot;an [[Ugly American (epithet)|ugly American]]&quot;, of greedy multinational corporations, and the [[military-industrial complex]]. They always introduced themselves with &quot;''beuuarhh''&quot; ({{IPA|fr|bœwaʁ|IPA}}), a slurred version of &quot;[[wikt:bonsoir|bonsoir]]&quot; ({{IPA|fr|bɔ̃.swaʁ|IPA}}, ''good evening''). During the first [[Gulf War]], the ''Guignols'' introduced a character called ''Commandant Sylvestre''. He would explain the war in broad oversimplified terms (&quot;''Here's the good guys, that's us, and here are the ragheads, so we'll kill everybody there...''&quot;). After the gulf war, he was reintroduced as Mr. Sylvestre, an ubiquitous executive from the military-industrial complex, the [[corporation|corporate]] world, and the [[CIA]], all mixed into the fictional mega-corporation ''World Company''. Sylvestre was dressed in a suit and tie, with a security badge. Other Sylvestres, dressed as Cardinals, Reverends, Imams, Rabbis, and other religious leaders, were also portrayed as the Church Company, the twin sister of the World Company, specializing in &quot;the business of religion&quot;. During the 2017 season, he was the show's main anchor. In the very last episode, he was the CEO who fired PPD and Jacques Chirac.

96 * '''[[George W. Bush]]''' was depicted as a cretin along with his father.&lt;ref name=&quot;:1&quot; /&gt; He shows a tendency to war and fights terrorism in his bedroom, defending himself with hand grenades (beer cans). His laptop password is &quot;connard&quot; (one of the French words for &quot;dumbass&quot;). He often appears along with one of the Sylvestres, who gets portrayed as the guy who's really in charge. Remarkably, Bush's character spoke in french with an american (or english, foreign) accent, whereas M. Sylvestre spoke french without any foreign accent, after the french dubbed voice of Sylvester Stallone in his films.

97 Steiger played [[Pontius Pilate]] in [[Franco Zeffirelli]]'s TV [[miniseries]] ''[[Jesus of Nazareth (TV series)|Jesus of Nazareth]]'' (1977). [[Stacy Keach]], who portrayed [[Barabbas]], expressed his joy at the opportunity to work with Steiger, describing him as &quot;generous and opinionated&quot;.{{sfn|Keach|2013|p=119}} In 1978, Steiger played a senator in Norman Jewison's ''[[F.I.S.T. (film)|F.I.S.T.]]'', opposite [[Sylvester Stallone]], who played a Cleveland warehouse worker involved in the labor union leadership of the fictional organisation named Federation of Inter-State Truckers.{{sfn|Greene|2010|p=109}} ''[[Love and Bullets (1979 film)|Love and Bullets]]'', later that year, in which Steiger appeared as a mafia boss, was poorly received; [[Roger Ebert]] dismissed it as a &quot;hopelessly confused hodgepodge of chases, killings, enigmatic meetings and separations, and insufferably overacted scenes by Steiger alternating with alarmingly underacted scenes by [Charles] Bronson&quot;.&lt;ref name=&quot;Ebert79&quot;&gt;{{cite web|url=http://www.rogerebert.com/reviews/love-and-bullets-1979|title=''Love and Bullets''|author=Ebert, Roger|publisher=RogerEbert.com|date=January 1, 1979|access-date=July 23, 2015|archive-url=https://web.archive.org/web/20150928062044/http://www.rogerebert.com/reviews/love-and-bullets-1979|archive-date=September 28, 2015|url-status=live}}&lt;/ref&gt; The following year, Steiger was cast as a general opposite Richard Burton and Robert Mitchum in [[Andrew V. McLaglen]]'s war film ''[[Breakthrough (1979 film)|Breakthrough]]'', set on the [[Western Front (World War II)|Western Front]].{{sfn|Bowker|2000|p=218}} In ''[[The Amityville Horror (1979 film)|The Amityville Horror]]'' (1979), Steiger appeared as a disturbed priest, who is invited to perform an [[exorcism]] on a haunted house. Again Steiger was accused of overacting; [[Janet Maslin]] of ''The New York Times'' wrote: &quot;Mr. Steiger bellows and weeps and overdoes absolutely everything. He won't even pick up the phone before it's rung 12 or 15 times.&quot;&lt;ref&gt;{{cite news|author=Maslin, Janet|url=https://www.nytimes.com/movie/review?res=9A07EFD6143EE732A25754C2A9619C946890D6CF|title=''The Amityville Horror'' (1979)|work=The New York Times|date=July 27, 1979|access-date=July 23, 2015|archive-url=https://web.archive.org/web/20160426064528/http://www.nytimes.com/movie/review?res=9A07EFD6143EE732A25754C2A9619C946890D6CF|archive-date=April 26, 2016 |url-status=live}}&lt;/ref&gt;

```
Pauline Kael thought that Steiger's &quot;spiritual agony was enough to shatter the camera lens&quot;.{{sfn|Hutchinson|1998|p=141}}
In 1993, Steiger portrayed an aging [[Gynaecology|gynaecologist]] who terrorizes his urban neighbors in a rural community in [[Burlington, Vermont]] in ''[[The Neighbor (1993 film)|The Neighbor]]''. Dennis Schwartz considered it to have been one of Steiger's creepiest roles, though he thought that the poor script had rendered the role awkward and &quot;mildly entertaining in the sense that Steiger is asked to carry the film and hams it up&quot;.&lt;ref name=&quot;Schwartz Neighbor&quot;&gt;{{cite web|author=Schwartz, Dennis|url=http://homepages.sover.net/~ozus/neighbor.htm|title=''The Neighbor''|publisher=Ozus' World Movie Reviews|date=February 20, 2004|access-date=July 23, 2015|archive-url=https://web.archive.org/web/20150928063826/http://homepages.sover.net/~ozus/neighbor.htm|archive-date=September 28, 2015|url-status=live}}&lt;/ref&gt; The following year, Steiger agreed to play the role of a Cuban mob boss opposite Sylvester Stallone and [[Sharon Stone]] in [[Luis Llosa]]'s thriller ''[[The Specialist]]'', citing its purpose as a &quot;$40 million commercial&quot; to show a new generation that he existed.{{sfn|Fantle|Johnson|2009|p=140}} Critics panned the film, which has a four percent approval rating on [[Rotten Tomatoes]] based on 27 reviews as of July 2015.&lt;ref&gt;{{cite web|url=http://www.rottentomatoes.com/m/1056079-specialist/|website=[[Rotten Tomatoes]]|access-date=July 23, 2015|title=''The Specialist'' (1994)|date=January 31, 2006 |archive-url=https://web.archive.org/web/20150928063845/http://www.rottentomatoes.com/m/1056079-specialist/|archive-date=September 28, 2015|url-status=live}}&lt;/ref&gt; The role earned Steiger a [[Golden Raspberry Award for Worst Supporting Actor]] nomination, and the film was listed in ''[[The Official Razzie Movie Guide]]'' as one of &quot;The 100 Most Enjoyably Bad Movies Ever Made&quot;.{{sfn|Wilson|2005}}
Leonard has also worked as an actor. He has appeared in numerous television shows, including ''[[Half &amp; Half]]'', ''[[L.A. Heat (TV series)|L.A. Heat]]'', ''[[Married... with Children]]'', ''[[Renegade (TV series)|Renegade]]'' and ''[[Tales From The Crypt (TV series)|Tales From The Crypt]]''. He has also appeared in several movies, including ''[[I Spy (2002 film)|I Spy]]'' and most recently ''[[The Fighter]]'' (2010), starring Christian Bale and Mark Wahlberg. This movie brought back memories of his fight with [[Dicky Eklund]]. He also worked as an adviser in the 2011 robot boxing film ''[[Real Steel]]''. Leonard served as host and mentor to the aspiring fighters on ''[[The Contender (TV series)|The Contender]]''. [[Sylvester Stallone]], who co-hosted during the first season, was one of the executive producers, along with [[Mark Burnett (executive producer)|Mark Burnett]]. When Leonard left the show, he was replaced as host by [[Tony Danza]] for the final season.&lt;ref&gt;{{IMDb title|0416359|The Contender}}&lt;/ref&gt;
    <title>Sylvester Stallone</title>
| name                = Sylvester Stallone
| image               = Sylvester Stallone Cannes 2019.jpg
| works               = [[Sylvester Stallone filmography|Filmography]]
| awards              = [[List of awards and nominations received by Sylvester Stallone|Full list]]
| signature           = Sylvester Stallone's signature.svg
* Sylvester Enzio Stallone&lt;ref name=&quot;bio&quot;&gt;{{Cite web|url=https://sylvesterstallone.com/bio/|title=Bio - About Sylvester Stallone|website=sylvesterstallone.com|access-date=January 19, 2023|archive-date=June 1, 2001|archive-url=https://web.archive.org/web/20010601150127/https://sylvesterstallone.com/bio/|url-status=live}}&lt;/ref&gt;
'''Sylvester Gardenzio Stallone''' ({{IPAc-en|s|t|ə|ˈ|l|oʊ|n}}; born July 6, 1946) is an American actor and filmmaker. He has received [[List of awards and nominations received by Sylvester Stallone|numerous accolades]], including a [[Golden Globe Awards|Golden Globe Award]] and a [[Critics' Choice Movie Awards|Critics' Choice Award]], as well as nominations for three [[Academy Awards]] and two [[British Academy Film Awards|BAFTA Awards]]. Stallone is one of only two actors in history (alongside [[Harrison Ford]]) to have starred in a box-office No. 1 film across six consecutive decades.&lt;ref&gt;{{cite web|url=https://sylvesterstallone.com/bio/|title=Bio-About Sylvester Stallone|website=sylvesterstallone.com|date=|access-date=June 23, 2021|archive-date=June 1, 2001|archive-url=https://web.archive.org/web/20010601150127/https://sylvesterstallone.com/bio/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://deadline.com/2021/08/sylvester-stallone-six-decades-number-one-film-rocky-rambo-the-suicide-squad-1234814038/|title=Sylvester Stallone Punches Way To KO'Ing Number One Film In Six Straight Decades|website=Deadline|date=August 12, 2021|access-date=August 13, 2021|archive-date=March 14, 2023|archive-url=https://web.archive.org/web/20230314181046/https://deadline.com/2021/08/sylvester-stallone-six-decades-number-one-film-rocky-rambo-the-suicide-squad-1234814038/|url-status=live}}&lt;/ref&gt;
Struggling as an actor for a number of years upon moving to [[New York City]] in 1969, Stallone found gradual work in films such as ''[[The Lords of Flatbush]]'' (1974). He achieved his greatest critical and commercial success starting in 1976 with his iconic role as boxer [[Rocky Balboa]] in the [[Rocky|first film]] of the successful [[Rocky (franchise)|''Rocky'' franchise]], which he also wrote.&lt;ref&gt;{{Cite web|title=Sylvester Stallone|url=http://www.imdb.com/name/nm0000230/|access-date=August 6, 2021|website=IMDb|archive-date=April 14, 2011|archive-url=https://web.archive.org/web/20110414005846/http://www.imdb.com/name/nm0000230/|url-status=live}}&lt;/ref&gt; In 1977, he became the third actor in history to be nominated for two Academy Awards for [[Academy Award for Best Original Screenplay|Best Original Screenplay]] and [[Academy Award for Best Actor|Best Actor]]. He portrayed the [[Post-traumatic stress disorder|PTSD]]-plagued soldier [[John Rambo]] in ''[[First Blood]]'' (1982), a role he would play across five [[Rambo (franchise)|''Rambo'' films]] (1982-2019). From the mid-1980s to the late 1990s, Stallone would go on to become one of Hollywood's highest-paid actors acting in action films such as ''[[Cobra (1986 film)|Cobra]]'' (1986), ''[[Tango &amp; Cash|Tango and Cash]]'' (1989), ''[[Cliffhanger (film)|Cliffhanger]]'' (1993), ''[[Demolition Man (film)|Demolition Man]]'' (1993), and ''[[The Specialist]]'' (1994). He received a star on the [[Hollywood Walk of Fame]] in 1984.
Sylvester Gardenzio Stallone&lt;ref name=&quot;digitalspy&quot;/&gt;&lt;ref name=&quot;Aging heroes&quot;/&gt;&lt;ref name=&quot;upi&quot;/&gt; was born in the [[Hell's Kitchen, Manhattan|Hell's Kitchen]] neighborhood of New York City's Manhattan borough&lt;ref name=guardianUK&gt;{{cite news|url=https://www.theguardian.com/film/2016/feb/24/sylvester-stallone-profile-creed-rocky|title=Sylvester Stallone: the wacky people's champ who battled his own ego|first=Ryan|last=Gilbey|work=[[The Guardian]]|location=UK|date=February 24, 2016|access-date=November 16, 2017|archive-date=April 8, 2023|archive-url=https://web.archive.org/web/20230408195719/https://www.theguardian.com/film/2016/feb/24/sylvester-stallone-profile-creed-rocky|url-status=live}}&lt;/ref&gt; on July 6, 1946,&lt;ref name=biography.com&gt;{{cite web|url=https://www.biography.com/people/sylvester-stallone-9491745|title=Sylvester Stallone: Director, Producer, Screenwriter, Actor, Film Actor (1946-)|publisher=[[Biography.com]] ([[FYI (TV network)|FYI]] / [[A&amp;E Networks]])|access-date=November 16, 2017|archive-date=December 6, 2017|archive-url=https://web.archive.org/web/20171206233506/https://www.biography.com/people/sylvester-stallone-9491745|url-status=live}}&lt;/ref&gt; the elder son of [[women's professional wrestling]] promoter [[Jackie Stallone|Jacqueline &quot;Jackie&quot; Stallone]] (née Labofish; 1921-2020) and hairdresser [[Frank Stallone Sr.|Francesco &quot;Frank&quot; Stallone Sr.]] (1919-2011).&lt;ref name=&quot;genealogy&quot;&gt;{{Cite web| language=it| url=https://storiedifamiglia.com/2023/09/05/dalla-puglia-agli-usa-gli-antenati-di-sylvester-stallone/| title=Dalla Puglia a gli USA! Gli antenati di Sylvester Stallone| website=storiedifamiglia.com| date=September 5, 2023| access-date=January 8, 2024| archive-date=January 8, 2024|archive-url=https://web.archive.org/web/20240108114141/https://storiedifamiglia.com/2023/09/05/dalla-puglia-agli-usa-gli-antenati-di-sylvester-stallone/| url-status=live}}&lt;/ref&gt; His mother was an American from [[Washington, D.C.]], with [[Bretons|Breton French]]&lt;ref&gt;[http://www.letelegramme.com/ig/generales/regions/bretagne/cinema-stallone-est-de-brest-meme-06-10-2009-596723.php &quot;Cinéma. Stallone est de Brest « même » !&quot;] {{Webarchive|url=https://web.archive.org/web/20091010001624/http://www.letelegramme.com/ig/generales/regions/bretagne/cinema-stallone-est-de-brest-meme-06-10-2009-596723.php |date=October 10, 2009 }} {{in lang|fr}}, ''Le Télégramme de Brest'', October 6, 2009&lt;/ref&gt; and [[History of the Jews in Ukraine|Ukrainian Jewish]] ancestry,&lt;ref&gt;{{cite web|last=Bloom|first=Nate|title=Interfaith Celebrities: Stallone's Jewish Grandfather|url=http://www.interfaithfamily.com/arts_and_entertainment/popular_culture/Interfaith_Celebrities_Stallones_Jewish_Grandfather.shtml|website=www.interfaithfamily.com|date=June 23, 2009|publisher=interfaith family|access-date=March 12, 2016|archive-date=June 26, 2009|archive-url=https://web.archive.org/web/20090626073935/http://www.interfaithfamily.com/arts_and_entertainment/popular_culture/Interfaith_Celebrities_Stallones_Jewish_Grandfather.shtml|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last1=Lightening|first1=L.M.|title=Sylvester Stallone's Jewish Mom Forced Him To Become A Hairdresser|url=http://jewishbusinessnews.com/2014/08/18/sylvester-stallones-jewish-mom-forced-him-to-become-a-hairdresser/|website=www.jewishbusinessnews.com|date=August 18, 2014|publisher=Jewish Business News|access-date=March 12, 2016|archive-date=April 12, 2015|archive-url=https://web.archive.org/web/20150412022311/http://jewishbusinessnews.com/2014/08/18/sylvester-stallones-jewish-mom-forced-him-to-become-a-hairdresser/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web| url=http://www.artukraine.com/old/historical/stallone.htm|archive-url=https://web.archive.org/web/20140809172653/http://www.artukraine.com/old/historical/stallone.htm| url-status=dead| archive-date=August 9, 2014| title=Mother of movie star Sylvester Stallone seeks grannie in Ukraine| website=ArtUkraine.com| access-date=July 5, 2015}}&lt;/ref&gt; while his father was an Italian immigrant from [[Gioia del Colle]]&lt;ref name=&quot;genealogy&quot; /&gt; who moved to the U.S. in the 1930s.&lt;ref&gt;{{cite web|url=https://www.youtube.com/watch?v=dsKu6TYaYpM |archive-url=https://ghostarchive.org/varchive/youtube/20211027/dsKu6TYaYpM| archive-date=October 27, 2021|title=Video of Stallone visiting Italy |work=Youtube | date=May 28, 2009|access-date=September 4, 2010}}{{cbignore}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://equiery.com/blog/?p=852|title=Polo Player and former Marylander Frank Stallone Dies|work=equiery.com|date=July 13, 2011|access-date=February 20, 2013|archive-date=March 23, 2016|archive-url=https://web.archive.org/web/20160323184634/http://equiery.com/blog/?p=852|url-status=live}}&lt;/ref&gt; His younger brother is actor and musician [[Frank Stallone]].&lt;ref name=&quot;genealogy&quot; /&gt; Many biographies of Stallone indicate that his birth name is &quot;Michael Sylvester Gardenzio Stallone&quot; and his mother explained in an interview that she originally named him &quot;Tyrone&quot; because she admired the actor [[Tyrone Power]], but Stallone's father changed it to &quot;Sylvester&quot;. His nickname as a child was &quot;Binky&quot; but he chose to go by the nickname of Mike/Michael after schoolmates began calling him &quot;Stinky&quot;.&lt;ref name=&quot;digitalspy&quot;&gt;{{Cite web|url=https://www.digitalspy.com/showbiz/10-things-about/a89634/ten-things-you-never-knew-about-sylvester-stallone/|title=Ten Things You Never Knew About Sylvester Stallone|website=digitalspy.com|publisher=[[Digital Spy]]|date=February 20, 2008|access-date=February 1, 2023|archive-date=January 20, 2023|archive-url=https://web.archive.org/web/20230120185911/https://www.digitalspy.com/showbiz/10-things-about/a89634/ten-things-you-never-knew-about-sylvester-stallone/|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Aging heroes&quot;&gt;{{Cite book| last1=Batchelor| first1=Bob| last2=Jones|first2=Norma| title=Aging heroes| url=https://books.google.com/books?id=OtNbCQAAQBAJ&amp;dq=%22Gardenzio%22&amp;pg=PA35| publisher=[[Rowman &amp; Littlefield]]| page=35| date=2015| isbn=978-1-4422-5006-2| access-date=March 19, 2023| archive-date=April 7, 2023| archive-url=https://web.archive.org/web/20230407174732/https://books.google.com/books?id=OtNbCQAAQBAJ&amp;dq=%22Gardenzio%22&amp;pg=PA35| url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;upi&quot;&gt;{{Cite web| url=https://www.upi.com/Archives/1987/12/17/TY-STALLONE/7459566715600| title=Ty Stallone?| website=upi.com|publisher=[[United Press International]]| date=December 17, 1987| access-date=February 1, 2023| archive-date=January 21, 2023|archive-url=https://web.archive.org/web/20230121162159/https://www.upi.com/Archives/1987/12/17/TY-STALLONE/7459566715600| url-status=live}}&lt;/ref&gt; His middle name &quot;Gardenzio&quot; is an alteration of the Italian given name &quot;[[Gaudenzio]]&quot; and he usually shortened it to &quot;Enzio&quot;.&lt;ref name=&quot;bio&quot; /&gt;
Complications during Stallone's birth forced his mother's obstetricians to use two pairs of [[forceps in childbirth|forceps]] while delivering him, accidentally severing a nerve in the process.&lt;ref&gt;{{cite web|author=The Biography Channel |title=Sylvester Stallone Biography |year=2007 |url=http://www.thebiographychannel.co.uk/biographies/sylvester-stallone.html |access-date=December 28, 2009 |url-status=dead |archive-url=https://web.archive.org/web/20091213163721/http://www.thebiographychannel.co.uk/biographies/sylvester-stallone.html |archive-date=December 13, 2009 }}&lt;/ref&gt;&lt;ref name=tca&gt;Stated on ''[[Inside the Actors Studio]]'', 1999&lt;/ref&gt; This caused paralysis of the lower left side of his face (including parts of his lip, tongue, and chin) which gave him his signature snarling look and slurred speech.&lt;ref name=tca/&gt;&lt;ref&gt;{{cite web|url=http://www.mid-day.com/articles/sylvester-stallone-can-still-pack-a-punch-at-68/15540185|title=Sylvester Stallone can still pack a punch at 68|date=August 21, 2014|work=mid-day|access-date=August 6, 2015|archive-date=April 27, 2019|archive-url=https://web.archive.org/web/20190427064858/https://www.mid-day.com/articles/sylvester-stallone-can-still-pack-a-punch-at-68/15540185|url-status=live}}&lt;/ref&gt; As a result, he was bullied in his childhood, with which he coped by getting into bodybuilding and acting.&lt;ref&gt;{{cite news|url=http://decoy.tvpassport.com/q_a/q-sylvester-stallone-has-very-interesting-voice-it-accent-or-something?subid=national-00001|title=Hollywood Q &amp; A|last=Thomlison|first=Adam|work=[[Winston-Salem Journal]]|date=August 21, 2021|accessdate=August 26, 2021|archive-date=August 14,
```



2022|archive-url=https://web.archive.org/web/20220814133615/http://decoy.tvpassport.com/q_a/q-sylvester-stallone-has-very-interesting-voice-it-accent-or-something?subid=national-00001|url-status=live}}&lt;/ref&gt; A guidance counselor once told Stallone's mother: "Your son is suited to run a sorting machine or to be an assistant electrician, primarily in the area of elevator operations&quot;; nevertheless, Sylvester aspired to be an actor and a screenwriter."&lt;ref&gt;{{Cite web |last=Güimil |first=Eva |date=2023-03-03 |title=Sylvester Stallone: How childhood rejection and bullying led the legendary actor to create 'Rocky' |url=https://english.elpais.com/culture/2023-03-03/sylvester-stallone-how-childhood-rejection-and-bullying-led-the-legendary-actor-to-create-rocky.html |access-date=2024-09-01 |website=EL PAÍS English |language=en-us}}&lt;/ref&gt;

Stallone spent part of his infancy in foster and boarding care, rejoining and moving back with his family to [[Maryland]] when he was five. In the early 1950s, his father moved the family to his mother's native Washington, D.C. to open a beauty school. In 1954, his mother opened a women's gym called Barbella's.&lt;ref&gt;Stallone, Sylvester. ''Sly Moves: My Proven Program to Lose Weight, Build Strength, Gain Will Power and Live Your Dream'', Rogue Marble Productions, 2005, p. 12.&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.usatoday.com/story/entertainment/celebrities/2020/09/22/jackie-stallone-dies-sylvester-stallone-mother-astrologist/5861878002/|title=Jackie Stallone, celebrity astrologer, Sylvester Stallone's eccentric mother, dies at 98|website=USA TODAY|access-date=May 20, 2023|archive-date=May 20, 2023|archive-url=https://web.archive.org/web/20230520151925/https://www.usatoday.com/story/entertainment/celebrities/2020/09/22/jackie-stallone-dies-sylvester-stallone-mother-astrologist/5861878002/|url-status=live}}&lt;/ref&gt; He initially stayed with his father following his parents' divorce when he was 11, but joined his remarried mother in [[Philadelphia]] when he was 15.&lt;ref&gt;{{cite news|url=https://www.britannica.com/biography/Sylvester-Stallone|title=Sylvester-Stallone|work=[[Britannica]]|date=April 2023|access-date=June 14, 2023|archive-date=March 15, 2023|archive-url=https://web.archive.org/web/20230315023235/https://www.britannica.com/biography/Sylvester-Stallone|url-status=live}}&lt;/ref&gt;

At one point, Stallone's teachers voted him &quot;Most Likely To End Up In The Electric Chair&quot;;&lt;ref&gt;{{Cite web |last=Shear |first=Robin |title=Celebrating UM Alumni |url=https://news.miami.edu/stories/2015/06/sharing-heart-and-humor.html |access-date=2024-09-01 |website=news.miami.edu |language=en}}&lt;/ref&gt; nevertheless, Stallone attended Notre Dame Academy and [[Abraham Lincoln High School (Philadelphia)|Abraham Lincoln High School]] in [[Philadelphia]],&lt;ref name=Edu&gt;{{cite web|last=Birnbaum|first=Aspen|title=Stallone, Sylvester (Sly)|url=http://pabook.libraries.psu.edu/palitmap/bios/Stallone__Sylvester.html|publisher=Pabook libraries|access-date=November 27, 2011|url-status=dead|archive-url=https://web.archive.org/web/20120114162642/http://pabook.libraries.psu.edu/palitmap/bios/Stallone__Sylvester.html|archive-date=January 14, 2012}}&lt;/ref&gt; and [[Charlotte Hall Military Academy]] in [[Charlotte Hall, Maryland]], prior to attending [[Miami Dade College]].&lt;ref name=&quot;Alumni&quot;&gt;{{Cite web|url=http://www.chma.org/listing.cfm?year=1964&amp;Submit=Submit|archive-url=https://web.archive.org/web/20111118103903/http://www.chma.org/listing.cfm?year=1964&amp;Submit=Submit|url-status=dead|title=Charlotte Hall Military Academy Alumni|archive-date=November 18, 2011}}&lt;/ref&gt; He spent two years, from September 1965 to June 1967, at the [[American College of Switzerland]]. He returned to the United States to study as a drama major at the [[University of Miami]], from 1967 to 1969.&lt;ref&gt;{{Cite web|url=https://sylvesterstallone.com/bio/|title=Bio &amp;#124; Sylvester Stallone|website=sylvesterstallone.com|access-date=June 14, 2019|archive-date=June 1, 2001|archive-url=https://web.archive.org/web/20010601150127/https://sylvesterstallone.com/bio/|url-status=live}}&lt;/ref&gt; After Stallone's request that his acting and life experiences be accepted in exchange for his remaining needed college credits to graduate, he was granted a [[Bachelor of Fine Arts]] (BFA) degree by the University of Miami in 1998.&lt;ref&gt;{{cite web|title=Celebrating UM Alumni|url=https://news.miami.edu/stories/2015/06/sharing-heart-and-humor.html|publisher=2015 University of Miami|access-date=March 25, 2023|archive-date=July 10, 2015|archive-url=https://web.archive.org/web/20150710062857/https://news.miami.edu/stories/2015/06/sharing-heart-and-humor.html|url-status=live}}&lt;/ref&gt;&lt;ref&gt; [https://www.themiamihurricane.com/2003/02/28/um-urban-legends-series-sylvester-stallones-um-porn-flick/ UM Urban Legends Series: Sylvester Stallone's UM porn flick.] {{Webarchive|url=https://web.archive.org/web/20230326043906/https://www.themiamihurricane.com/2003/02/28/um-urban-legends-series-sylvester-stallones-um-porn-flick/ |date=March 26, 2023 }} The Miami Hurricane. February 28, 2003. Retrieved March 25, 2023.&lt;/ref&gt;

}} Stallone: &quot;I was broke and basically sleeping in the Port Authority bus station for three weeks straight. I read in a trade paper about this film [''The Party at Kitty and Studs'', 1970] that was paying {{US$|100}} a day - for a {{US$|100}} a day I would wreak havoc. Instead of doing something desperate, I worked for two days for {{US$|200}} and got myself out of the bus station.&quot;&lt;/ref&gt; Stallone later explained that he had done the film out of desperation after being evicted from his apartment and finding himself homeless for several days. He has also said that he slept three weeks in the [[Port Authority Bus Terminal]] in New York City prior to seeing a casting notice for the film. In the actor's words, &quot;it was either do that movie or rob someone, because I was at the end - the very end - of my rope&quot;.&lt;ref&gt;[http://www.playboy.com/arts-entertainment/features/sylvester-stallone/sylvester-stallone-04.html Sylvester Stallone interview] {{webarchive|url=https://web.archive.org/web/20110606223229/http://www.playboy.com/arts-entertainment/features/sylvester-stallone/sylvester-stallone-04.html |date=June 6, 2011 }}, ''[[Playboy (magazine)|Playboy]]'', September 1978.&lt;/ref&gt; The film was released several years later as ''Italian Stallion'', in order to cash in on Stallone's newfound fame (the new title was taken from Stallone's nickname since ''Rocky''). Stallone also starred in the erotic [[off-Broadway]] stage play ''Score'' which ran for 23 performances at the Martinique Theatre from October 28 to November 15, 1971, and was later made into the 1974 film ''[[Score (1974 film)|Score]]'' by [[Radley Metzger]].&lt;ref name=&quot;FM-Summer1997&quot;&gt;{{citation|last=Gallagher|first=Steve|title=The Libertine|url=http://filmmakermagazine.com/archives/issues/summer1997/metzger.php|date=Summer 1997|work=[[Filmmaker (magazine)|Filmmaker Magazine]]|access-date=May 24, 2015|archive-date=September 22, 2015|archive-url=https://web.archive.org/web/20150922002127/http://filmmakermagazine.com/archives/issues/summer1997/metzger.php|url-status=live}}&lt;/ref&gt;

After moving to New York City, Stallone shared an apartment with his girlfriend, Sasha Czack, an aspiring actress who supported them by working as a waitress.&lt;ref&gt;{{Cite web|url=https://www.sheknows.com/entertainment/articles/1128921/sylvester-stallones-life-timeline/|title=Sylvester Stallone's taken more punches in life than he has on the big screen|first=Lauren|last=Kelly|website=She Knows|date=October 11, 2016|access-date=March 30, 2020|archive-date=October 12, 2016|archive-url=https://web.archive.org/web/20161012142330/https://www.sheknows.com/entertainment/articles/1128921/sylvester-stallones-life-timeline/|url-status=live}}&lt;/ref&gt; Stallone took odd jobs around this time, including being a cleaner at a zoo, and a theater usher; he was fired from the latter for scalping tickets. He furthered his writing skills by frequenting a local library, and became interested in the works of [[Edgar Allan Poe]].&lt;ref&gt;{{Cite web|url=https://www.looper.com/27729/truth-sylvester-stallone/|title=The truth about Sylvester Stallone|first=Brian|last=Boone|date=October 13, 2016|website=[[Looper.com]]|access-date=March 30, 2020|archive-date=October 17, 2016|archive-url=https://web.archive.org/web/20161017015946/https://www.looper.com/27729/truth-sylvester-stallone/|url-status=live}}&lt;/ref&gt;

In 1972, Stallone was on the verge of giving up on having an acting career; in what he later described as a low point, he tried and failed to get a job as an extra in ''[[The Godfather]]''.&lt;ref&gt;{{Cite web|url=https://indianexpress.com/article/entertainment/hollywood/sylvester-stallone-was-rejected-as-extra-in-the-godfather/|title=Sylvester Stallone was rejected as extra in 'The Godfather'|newspaper=The Indiana Express|date=February 27, 2016|access-date=March 30, 2020|archive-date=February 28, 2016|archive-url=https://web.archive.org/web/20160228112725/https://indianexpress.com/article/entertainment/hollywood/sylvester-stallone-was-rejected-as-extra-in-the-godfather/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.projectcasting.com/news/sylvester-stallone-the-godfather/|title=Stallone: I Was Turned Down for an Extra Role in 'The Godfather'|date=February 25, 2016|access-date=March 30, 2020|archive-date=February 27, 2016|archive-url=https://web.archive.org/web/20160227024621/https://www.projectcasting.com/news/sylvester-stallone-the-godfather/|url-status=live}}&lt;/ref&gt; Instead, he was relegated to a background role in another Hollywood hit, ''[[What's Up, Doc? (1972 film)|What's Up, Doc?]]'', starring [[Barbra Streisand]]. Stallone is hardly visible in his two appearances.

[[File:Sylvester Stallone - 1977.jpg|thumb|upright|Stallone at the [[Ken Norton#Norton vs. Young|Ken Norton vs. Duane Bobick boxing match]] in 1977]]

Stallone occasionally attempted, albeit unsuccessfully, roles in different genres. In 1984, he co-wrote and starred alongside [[Dolly Parton]] in the comedy film ''[[Rhinestone (film)|Rhinestone]]'', where he played a wannabe country music singer. For the [[Rhinestone (film soundtrack)|''Rhinestone'' soundtrack]], he performed a song. Stallone turned down the lead male role in ''[[Romancing the Stone]]'' in order to make ''Rhinestone'' instead, a decision he later regretted.&lt;ref&gt;{{cite magazine|date=January 14, 2014|title=Sylvester Stallone: in his own words|url=https://www.empireonline.com/movies/features/evening-sylvester-stallone-quotes/|magazine=Empire|access-date=March 31, 2018|archive-date=May 1, 2018|archive-url=https://web.archive.org/web/20180501010256/https://www.empireonline.com/movies/features/evening-sylvester-stallone-quotes/|url-status=live}}&lt;/ref&gt;

In 1985, Stallone continued his success with the ''Rocky'' and ''Rambo'' franchises with ''[[Rocky IV]]'' and ''[[Rambo: First Blood Part II]].'' Stallone has portrayed these two characters in a total of 13 films. Stallone met former [[Mr. Olympia]] [[Franco Columbu]] to develop his character's appearance for the film ''Rocky IV'', just as if he were preparing for the Mr. Olympia competition. That meant two workouts a day, six days a week.&lt;ref&gt;{{Cite web|url=http://sylvesterstallone.com/health-fitness/four/|archive-url=https://web.archive.org/web/20120905145716/http://sylvesterstallone.com/health-fitness/four/|url-status=dead|title=Sylvester Stallone - Four&lt;!-- Bot generated title --&gt;|archive-date=September 5, 2012}}&lt;/ref&gt; Both films were major financial successes.

In 1987, he starred in the family drama ''[[Over the Top (1987 film)|Over the Top]]'' as a struggling trucker who tries to make amends with his estranged son and enters an [[arm wrestling]] competition. This was poorly received by critics and was a box-office failure.&lt;ref&gt;{{Cite web|last=Bailey|first=Jason|date=July 8, 2013|title=So Bad It's Good: Sylvester Stallone, Arm Wrestling, and '80s Nostalgia in 'Over the Top'|url=https://www.flavorwire.com/402226/so-bad-its-good-sylvester-stallone-arm-wrestling-and-80s-nostalgia-in-over-the-top|access-date=July 5, 2021|website=Flavorwire|archive-date=July 20, 2021|archive-url=https://web.archive.org/web/20210720102726/https://www.flavorwire.com/402226/so-bad-its-good-sylvester-stallone-arm-wrestling-and-80s-nostalgia-in-over-the-top|url-status=live}}&lt;/ref&gt; In 1989, he co-starred alongside [[Kurt Russell]] in the [[buddy cop]] action film ''[[Tango &amp; Cash]]'', which did solid business domestically and overseas, grossing {{US$|57}}&amp;nbsp;million in foreign markets and over {{US$|120}}&amp;nbsp;million worldwide.&lt;ref&gt;{{Cite web|url=http://jpbox-office.com/fichfilm.php?id=5531|title=Tango &amp; Cash (1989) - JPBox-Office|website=jpbox-office.com|access-date=April 21, 2021|archive-date=April 21, 2021|archive-url=https://web.archive.org/web/20210421153142/http://jpbox-office.com/fichfilm.php?id=5531|url-status=live}}&lt;/ref&gt; Stallone became a boxing promoter in the 1980s. His boxing promoting company, Tiger Eye Productions, signed world champion boxers [[Sean O'Grady (boxer)|Sean O'Grady]] and [[Aaron Pryor]].&lt;ref&gt;{{cite web|title=Sylvester Stallone|url=http://boxrec.com/media/index.php/Sylvester_Stallone|website=boxrec.com|access-date=September 8, 2014|archive-date=April 18, 2015|archive-url=https://web.archive.org/web/20150418131744/http://boxrec.com/media/index.php/Sylvester_Stallone|url-status=live}}&lt;/ref&gt;

[[file:Sylvester Stallone Cannes.jpg|thumb|upright|Stallone at the 1993 [[Cannes Film Festival]]]]

In 1996, he starred in the [[disaster film]] ''[[Daylight (1996 film)|Daylight]]'' as a disgraced former emergency services chief who attempts to rescue survivors of an underground tunnel explosion. ''Daylight'' also underperformed at the domestic box office, grossing $33 million, but did better overseas and grossed a total of $158 million worldwide.&lt;ref&gt;{{Cite web|last=Gomez|first=Patrick|date=July 2, 2020|title=This Sylvester Stallone flop still makes for an explosive good time|url=https://www.avclub.com/this-sylvester-stallone-flop-still-makes-for-an-explosi-1844191264|access-date=July 5, 2021|website=The A.V. Club|archive-date=July 9, 2021|archive-url=https://web.archive.org/web/20210709183357/https://www.avclub.com/this-sylvester-stallone-flop-still-makes-for-an-explosi-1844191264|url-status=live}}&lt;/ref&gt; In 1997, Stallone was cast against type as an overweight sheriff in the crime drama ''[[Cop Land]]'', in which he starred alongside [[Robert De Niro]] and [[Ray Liotta]]. The film was critically well-received and was a modest success at the box office, earning $63 million on a $15 million budget, and Stallone's performance earned him the [[Stockholm International Film Festival]] Best Actor Award. In 1998, he did voice-over work for the animated film ''[[Antz]]'', which was a success domestically.

In 2003, he played a villainous role in the third installment of the ''[[Spy Kids (film series)|Spy Kids]]'' series: ''[[Spy Kids 3-D: Game Over]]'', which was a huge box-office success (almost {{US$|200}}&amp;nbsp;million worldwide). Stallone also had a cameo appearance in the 2003 French film ''[[Taxi 3]]'' as a passenger. Also that year, Stallone started to regain prominence for his supporting role in the [[neo-noir]] crime drama ''[[Shade (film)|Shade]]'' which was only released in a limited fashion but was praised by critics.&lt;ref&gt;{{cite web |url=https://www.rottentomatoes.com/m/shade/ |title=Shade |website=Rotten Tomatoes |date=July 21, 2009 |access-date=September 4, 2010 |archive-date=July 28, 2010

|archive-url=https://web.archive.org/web/20100728052059/http://www.rottentomatoes.com/m/shade/ |url-status=live }}&lt;/ref&gt; He was also attached to star and direct a film tentatively titled ''Rampart Scandal'', which was to be about the murder of rappers [[Tupac Shakur]] and [[The Notorious B.I.G.]] and the surrounding [[Los Angeles Police Department]] corruption scandal.&lt;ref&gt;{{cite news|url=http://www.mtv.com/news/1472396/sylvester-stallone-making-movie-about-biggie-tupac-murders/|title=Sylvester Stallone Making Movie About Biggie, Tupac Murders|last=Patel|first=Joseph|date=June 6, 2003|publisher=[[MTV News]]|access-date=June 9, 2020|archive-date=June 6, 2020|archive-url=https://web.archive.org/web/20200606204428/http://www.mtv.com/news/1472396/sylvester-stallone-making-movie-about-biggie-tupac-murders/|url-status=dead}}&lt;/ref&gt; It was later titled ''Notorious'' but was shelved.&lt;ref&gt;{{cite web|url=http://www.eurweb.com/story/eur30196.cfm |title=Stallone's Tupac/Biggie Movie a No Go: Actor was to play LAPD detective who found dirty cops at root of murders |date=December 7, 2006 |publisher=EURWeb.com |access-date=January 9, 2010 }}{{dead link|date=June 2016|bot=medic}}{{cbignore|bot=medic}}&lt;/ref&gt;

In 2005, alongside [[Sugar Ray Leonard]], he was the co-presenter of the [[NBC]] reality television boxing competition series ''[[The Contender (TV series)|The Contender]]''. That same year he also made a guest appearance in two episodes of the television series ''[[Las Vegas (TV series)|Las Vegas]]''. That year, Stallone also inducted [[professional wrestling]] icon [[Hulk Hogan]], who appeared in ''Rocky III'' as a wrestler named Thunderlips, into the WWE Hall of Fame; Stallone was also the person who offered Hogan the cameo in ''Rocky III''.&lt;ref&gt;{{cite web |url=http://www.donaly.com/celebrity_scene_weekly.html |title=Sylvester Stallone Rocky- Celebrity Scene Monthly By Don Aly Vol 36 |publisher=Donaly.com |date=August 19, 2010 |access-date=September 4, 2010 |archive-date=July 21, 2010 |archive-url=https://web.archive.org/web/20100721050133/http://www.donaly.com/celebrity_scene_weekly.html |url-status=dead }}&lt;/ref&gt; In August, Stallone released his book ''Sly Moves'' which claimed to be a guide to fitness and nutrition as well as a candid insight into his life and works from his own perspective. The book also contained many photographs of Stallone throughout the years as well as pictures of him performing exercises.

[[file:Sylvester Stallone Rocky VI 2005.JPG|thumb|16 years after filming ''[[Rocky V]]'', Stallone reprised his role as [[Rocky Balboa]] in 2006.]]

After a three-year hiatus from films, Stallone had a comeback in 2006 with the sixth installment of the ''Rocky'' series, ''[[Rocky Balboa (film)|Rocky Balboa]]'', which was a critical and commercial hit. After the critical and box-office failure of the previous installment ''[[Rocky V]]'', Stallone had decided to write, direct and star in a sixth installment which would be a more appropriate climax to the series. The total domestic box office came to {{US$|70}}.3&amp;nbsp;million (and {{US$|155.7}}&amp;nbsp;million worldwide).&lt;ref&gt;{{cite web |url=https://boxofficemojo.com/movies/?id=rocky6.htm |title=Rocky Balboa |website=[[Box Office Mojo]] |access-date=September 4, 2010 |archive-date=August 19, 2010 |archive-url=https://web.archive.org/web/20100819053715/http://www.boxofficemojo.com/movies/?id=rocky6.htm |url-status=live }}&lt;/ref&gt; The budget of the film was only {{US$|24}}&amp;nbsp;million. His performance in ''Rocky Balboa'' has been praised and garnered mostly positive reviews.&lt;ref&gt;{{cite web|url=https://www.rottentomatoes.com/m/rocky_balboa/|title=Rocky Balboa|date=December 22, 2006|work=rottentomatoes.com|access-date=February 19, 2020|archive-date=November 17, 2020|archive-url=https://web.archive.org/web/20201117080022/https://www.rottentomatoes.com/m/rocky_balboa/|url-status=live}}&lt;/ref&gt; That year, the development [[Death Wish (2018 film)|''Death Wish'']] remake began, when Stallone announced that he would be directing and starring in a remake of [[Death Wish (1974 film)|the 1974 film]]. Stallone said, &quot;Instead of the [[Charles Bronson]] character being an architect, my version would have him as a very good cop who had incredible success without ever using his gun. So when the attack on his family happens, he's really thrown into a moral dilemma in proceeding to carry out his revenge.&quot; He later told the publication that he was no longer involved.&lt;ref name=&quot;MHoleRemake&quot;&gt;{{cite news|last=Morris|first=Clint|date=June 8, 2008|title=Stallone in Death Wish remake? Updated!|work=Moviehole|url=http://www.moviehole.net/news/20061220_stallone_in_death_wish_remake.html|url-status=dead|access-date=December 20, 2006|archive-url=https://web.archive.org/web/20070703060619/http://www.moviehole.net/news/20061220_stallone_in_death_wish_remake.html|archive-date=July 3, 2007|quote=Upon listening to the talkback responses on AICN, many who turned their nose at the idea of a remake, Stallone tells the site today that he will NOT be doing the movie. Yep, he listened to the fans!}}&lt;/ref&gt;&lt;ref&gt;{{cite news|author=&lt;!--Not stated--&gt;|date=November 5, 2007|title=Stallone tapped for Death Wish remake|work=[[The Guardian]]|url=http://film.guardian.co.uk/news/story/0,,2205587,00.html|access-date=November 5, 2007|archive-date=November 6, 2007|archive-url=https://web.archive.org/web/20071106100219/http://film.guardian.co.uk/news/story/0,,2205587,00.html|url-status=live}}&lt;/ref&gt; In a 2009 interview with [[MTV]], though, Stallone stated that he was again considering the project.&lt;ref&gt;{{cite web|last1=Rosenberg|first1=Adam|date=October 2, 2009|title=Sylvester Stallone Speaks on a 'Death Wish' Remake and Edgar Allen Poe|url=http://www.mtv.com/news/2433850/sylvester-stallone-speaks-on-a-death-wish-remake-and-edgar-allen-poe/|access-date=February 21, 2018|work=[[MTV News]]|publisher=[[MTV]]|archive-date=February 21, 2018|archive-url=https://web.archive.org/web/20180221100840/http://www.mtv.com/news/2433850/sylvester-stallone-speaks-on-a-death-wish-remake-and-edgar-allen-poe/|url-status=dead}}&lt;/ref&gt; However the role went to [[Bruce Willis]] with [[Eli Roth]] as director.

Stallone partnered with a beverage company producing an upscale [[bottled water]] brand called Sly Water.&lt;ref&gt;{{cite web|author=Dietrich, Heidi|date=January 15, 2006|title=Sylvester Stallone taps Mount Rainier for water sales|url=http://www.bizjournals.com/seattle/stories/2006/01/16/story2.html|access-date=December 6, 2015|work=Puget Sound Business Journal|archive-date=May 9, 2007|archive-url=https://web.archive.org/web/20070509061112/http://www.bizjournals.com/seattle/stories/2006/01/16/story2.html|url-status=live}}&lt;/ref&gt;

In July 2009, Stallone made a cameo appearance in the [[Bollywood]] film ''[[Kambakkht Ishq]]'', where he played himself.&lt;ref&gt;{{cite web|date=February 22, 2010|title=Sylvester Stallone And Denise Richards Nominated For Razzies Equivalent, The Golden Kela Awards|url=http://www.mtv.com/news/2435621/sylvester-stallone-and-denise-richards-nominated-for-razzies-equivalent-the-golden-kela-awards/|url-status=dead|archive-url=https://web.archive.org/web/20100224081228/http://moviesblog.mtv.com/2010/02/22/sylvester-stallone-and-denise-richards-nominated-for-razzies-equivalent-the-golden-kela-awards/|archive-date=February 24, 2010|access-date=June 9, 2020|publisher=MTV}}&lt;/ref&gt;

[[File:Flickr - nicogenin - 66ème Festival de Venise (Mostra) - Sylvester Stallone (26).jpg|thumb|upright|left|Stallone in 2009 at the [[66th Venice International Film Festival]]]]

It was announced on December 7, 2010, that Stallone was voted into the [[International Boxing Hall of Fame]] in the non-participant category.&lt;ref&gt;{{cite news|date=December 7, 2010|title=Sylvester Stallone, hall of famer|work=Newsday|url=http://www.newsday.com/entertainment/pet-rock-1.811972/sylvester-stallone-hall-of-famer-1.2523881|url-status=dead|access-date=December 7, 2010|archive-url=https://web.archive.org/web/20101211122730/http://www.newsday.com/entertainment/pet-rock-1.811972/sylvester-stallone-hall-of-famer-1.2523881|archive-date=December 11, 2010}}&lt;/ref&gt;

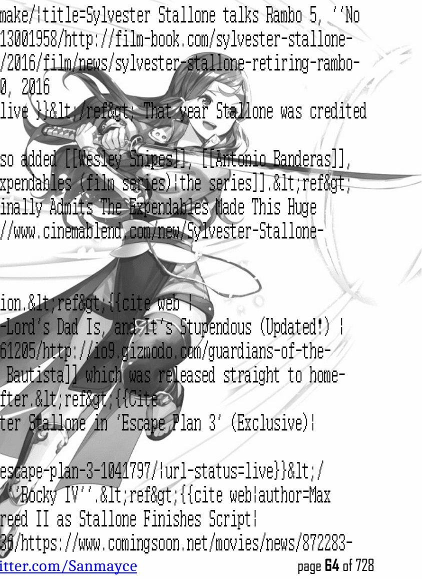
Also that year, Stallone wrote, directed and starred in the ensemble action film ''[[The Expendables (2010 film)|The Expendables]]''. The film, which was filmed during summer/winter 2009, was released on August 13, 2010. Joining him in the film were fellow action stars [[Jason Statham]], [[Jet Li]], and [[Dolph Lundgren]], as well as [[Terry Crews]], [[Mickey Rourke]], [[Randy Couture]], [[Eric Roberts]], and [[Stone Cold Steve Austin]], and cameos by fellow '80s action icons [[Bruce Willis]] and [[Arnold Schwarzenegger]].&lt;ref&gt;{{cite web |url=https://www.slashfilm.com/2009/07/09/action-movie-sequel-time-the-expendables-2-and-more-inglorious-basterds-prequel-talk/ |archive-url=https://web.archive.org/web/20090712030119/http://www.slashfilm.com/2009/07/09/action-movie-sequel-time-the-expendables-2-and-more-inglorious-basterds-prequel-talk/ |url-status=dead |archive-date=July 12, 2009 |title=Action Movie Sequel Time: The Expendables 2, And More Inglorious Basterds Prequel Talk |publisher=Slashfilm.com |date=July 9, 2009 |access-date=September 4, 2010 }}&lt;/ref&gt; The film took {{US$|34,825,135}} in its opening weekend, going straight in at No. 1 in the US box office. The figure marked the biggest opening weekend in Stallone's career.&lt;ref&gt;{{cite web |url=https://boxofficemojo.com/news/?id=2889&amp;p=.htm |title=Weekend Report: 'Expendables' Pump Up, 'Eat Pray Love' Pigs Out, 'Scott Pilgrim' Powers Down |website=Box Office Mojo |access-date=September 4, 2010 |archive-date=September 6, 2010 |archive-url=https://web.archive.org/web/20100906040308/http://www.boxofficemojo.com/news/?id=2889&amp;p=.htm |url-status=live }}&lt;/ref&gt; In summer 2010, Brazilian company O2 Filmes released a statement saying it was still owed more than {{US$|2}}&amp;nbsp;million for its work on the film.&lt;ref&gt;{{cite news |first= Tom |last= Phillips |location= Rio de Janeiro |url= https://www.theguardian.com/film/2010/aug/02/sylvester-stallone-pursued-brazilians-debts |title= Sylvester Stallone pursued by Brazilian company for unexpendable debts |work= The Guardian |date= August 2, 2010 |access-date= August 13, 2010 |archive-date= January 28, 2019 |archive-url= https://web.archive.org/web/20190128232902/https://www.theguardian.com/film/2010/aug/02/sylvester-stallone-pursued-brazilians-debts |url-status= live }}&lt;/ref&gt; In 2011, Stallone provided the voice of a lion in [[Kevin James]]' comedy ''[[Zookeeper (film)|Zookeeper]]''.

In 2013, Stallone starred in the action film ''[[Bullet to the Head]]'', directed by [[Walter Hill (director)|Walter Hill]], based upon Alexis Nolent's French graphic novel ''Du Plomb Dans La Tete''.&lt;ref&gt;{{cite web|url=http://sylvesterstallone.com/bullet-to-the-head-wraps-filming/|title=Bullet to the Head wraps filming|work=SylvesterStallone.com|date=August 29, 2011|access-date=July 24, 2012|archive-date=August 4, 2012|archive-url=https://web.archive.org/web/20120804084948/http://sylvesterstallone.com/bullet-to-the-head-wraps-filming/|url-status=live}}&lt;/ref&gt; Also in 2013, he starred in the action thriller ''[[Escape Plan (film)|Escape Plan]]'', along with Arnold Schwarzenegger and [[Jim Caviezel]], and in the sports comedy drama ''[[Grudge Match]]'' alongside [[Robert De Niro]], harkening back to the ''Rocky'' franchise. Stallone was reported to be developing an English-language remake of the Spanish film ''[[No Rest for the Wicked (film)|No Rest for the Wicked]]'', though the project was shelved.&lt;ref&gt;{{cite web|url=http://film-book.com/sylvester-stallone-talks-rambo-5-no-rest-for-the-wicked-remake/|title=Sylvester Stallone talks Rambo 5, ''No Rest For the Wicked'' Remake|date=March 9, 2012|access-date=July 24, 2012|archive-date=March 13, 2012|archive-url=https://web.archive.org/web/20120313001958/http://film-book.com/sylvester-stallone-talks-rambo-5-no-rest-for-the-wicked-remake/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine |author=Ramin Setoodeh |url=https://variety.com/2016/film/news/sylvester-stallone-retiring-rambo-1201672291 |title=Sylvester Stallone Retiring 'Rambo' |magazine=Variety |date=January 5, 2016 |access-date=December 8, 2016 |archive-date=December 20, 2016 |archive-url=https://web.archive.org/web/20161220214116/http://variety.com/2016/film/news/sylvester-stallone-retiring-rambo-1201672291/ |url-status=live }}&lt;/ref&gt; That year Stallone was credited as writer for the [[Jason Statham]] action film vehicle ''[[Homefront (2013 film)|Homefront]].''

''[[The Expendables 3]]'', the third installment in the ensemble action film series, was released on August 15, 2014. The returning ensemble cast also added [[Wesley Snipes]], [[Antonio Banderas]], [[Mel Gibson]] and [[Harrison Ford]]. This film was negatively received by both critics and audiences and became the lowest-grossing film in [[The Expendables (film series)|the series]].&lt;ref&gt;{{cite web|url=https://www.cinemablend.com/new/Sylvester-Stallone-Finally-Admits-Expendables-Made-Huge-Mistake-68375.html|title=Sylvester Stallone Finally Admits The Expendables Made This Huge Mistake|date=November 24, 2014|access-date=November 23, 2018|archive-date=May 24, 2020|archive-url=https://web.archive.org/web/20200524095457/https://www.cinemablend.com/new/Sylvester-Stallone-Finally-Admits-Expendables-Made-Huge-Mistake-68375.html|url-status=live}}&lt;/ref&gt;

[[File:Sylvester Stallone Cannes 2014 2.jpg|thumb|upright|Stallone promoting ''[[The Expendables 3]]'' at the [[2014 Cannes Film Festival]]]]

In 2017, Stallone appeared in ''[[Guardians of the Galaxy Vol. 2]]'' as [[Starhawk (comics)|Stakar Ogord / Starhawk]], the leader of a Ravagers faction.&lt;ref&gt;{{cite web | url=http://io9.gizmodo.com/guardians-of-the-galaxy-vol-2-reveals-who-star-lords-d-1784197253 | title=Guardians of the Galaxy Vol. 2 Reveals Who Star-Lord's Dad Is, and It's Stupendous (Updated!) | publisher=[[io9]] | date=July 23, 2016 | access-date=August 3, 2016 | archive-date=July 24, 2016 | archive-url=https://web.archive.org/web/20160724061205/http://io9.gizmodo.com/guardians-of-the-galaxy-vol-2-reveals-who-star-lords-d-1784197253 | url-status=live }}&lt;/ref&gt; In 2018, he co-starred in ''[[Escape Plan 2: Hades]]'' with [[Dave Bautista]] which was released straight to home-video. Upon wrapping production, he announced via his social media page that work on ''[[Escape Plan 3: Devil's Station]]'' began immediately thereafter.&lt;ref&gt;{{Cite web|url=https://www.hollywoodreporter.com/movies/movie-news/jaime-king-joins-sylvester-stallone-escape-plan-3-1041797/|title=Jaime King Joins Sylvester Stallone in 'Escape Plan 3' (Exclusive)|first1=Borys|last1=Kit|website=[[The Hollywood Reporter]]|date=September 20, 2017|access-date=October 24, 2021|archive-date=October 24, 2021|archive-url=https://web.archive.org/web/20211024180216/https://www.hollywoodreporter.com/movies/movie-news/jaime-king-joins-sylvester-stallone-escape-plan-3-1041797/|url-status=live}}&lt;/ref&gt; In July, Stallone announced that he had finished a script for a sequel to ''Creed'', with a plot including the return of [[Ivan Drago]] from ''Rocky IV''.&lt;ref&gt;{{cite web|author=Max Evry|url=https://www.comingsoon.net/movies/news/872283-drago-confirmed-for-creed-ii-as-stallone-finishes-script#/slide/1|title=Drago Confirmed for Creed II as Stallone Finishes Script|publisher=ComingSoon|date=July 21, 2017|access-date=July 21, 2017|archive-date=November 9, 2020|archive-url=https://web.archive.org/web/20201109014836/https://www.comingsoon.net/movies/news/872283-



drago-confirmed-for-creed-ii-as-stallone-finishes-script#/slide/1|url-status=live}}&lt;/ref&gt; That year, Stallone was featured in [[Derek Wayne Johnson]]'s ''John G. Avildsen: King of the Underdogs'', a documentary about director [[John G. Avildsen]].&lt;ref&gt;Kreps, Daniel. [https://www.rollingstone.com/movies/movie-news/john-g-avildsen-rocky-the-karate-kid-director-dead-at-81-199305/ &quot;John G. Avildsen, ''Rocky'', ''The Karate Kid'' Director, Dead at 81&quot;] {{Webarchive|url=https://web.archive.org/web/20190415050025/https://www.rollingstone.com/movies/movie-news/john-g-avildsen-rocky-the-karate-kid-director-dead-at-81-199305/ |date=April 15, 2019 }}, ''Rolling Stone'', San Francisco, CA, June 17, 2017. Retrieved on August 21, 2018.&lt;/ref&gt;

135 ''[[Creed II]]'' went into production in March 2018, with a scheduled release on Thanksgiving 2018. Stallone was originally slated to direct before the appointment of [[Steven Caple Jr.]], in his feature film directorial debut.&lt;ref&gt;{{cite web|url=http://ftw.usatoday.com/2017/12/creed-2-release-date-sylvester-stallone-ivan-drago|title=Sylvester Stallone drops major {{sic|announc|ment|nolink=y}} about 'Creed II'|date=December 27, 2017|access-date=December 31, 2017|archive-date=December 31, 2017|archive-url=https://web.archive.org/web/20171231155408/http://ftw.usatoday.com/2017/12/creed-2-release-date-sylvester-stallone-ivan-drago|url-status=live}}&lt;/ref&gt; ''Creed II'' was released in the United States by [[Metro-Goldwyn-Mayer]] on November 21, 2018. The film received generally positive reviews from critics and it went on to debut to $35.3 million in its opening weekend (a five-day total of $55.8 million), marking the biggest debut ever for a live-action release over Thanksgiving.&lt;ref&gt;{{cite web|last=Campbell|first=Christopher|title=Creed II First Reviews: A Solid, if Predictable, Sequel|url=https://editorial.rottentomatoes.com/article/creed-ii-first-reviews-a-solid-if-predictable-sequel/|work=[[Rotten Tomatoes]]|access-date=November 16, 2018|date=November 18, 2018|archive-date=April 23, 2019|archive-url=https://web.archive.org/web/20190423072752/http://editorial.rottentomatoes.com/article/creed-ii-first-reviews-a-solid-if-predictable-sequel/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.forbes.com/sites/scottmendelson/2018/11/25/michael-b-jordans-creed-2-kos-box-office-with-record-breaking-55m-debut/|title=Michael B. Jordan's 'Creed 2' KO's Box Office With Record-Breaking $55M Debut|website=[[Forbes]]|access-date=November 25, 2018|archive-date=November 25, 2018|archive-url=https://web.archive.org/web/20181125182651/https://www.forbes.com/sites/scottmendelson/2018/11/25/michael-b-jordans-creed-2-kos-box-office-with-record-breaking-55m-debut/|url-status=live}}&lt;/ref&gt;

136 On July 23, 2019, in an interview with [[Variety (magazine)|Variety]], Stallone said that a Rocky sequel and prequel are in development. Producer [[Irwin Winkler]] said &quot;We're very high on it&quot; and that negotiations are underway for Stallone to write and star in the feature. &quot;We're very anxious to make it.&quot; Stallone said the plot of the film would be about Rocky befriending a young fighter who is an undocumented immigrant. &quot;Rocky meets a young, angry person who got stuck in this country when he comes to see his sister. He takes him into his life, and unbelievable adventures begin, and they wind up south of the border. It's very, very timely.&quot; Stallone said. Stallone also said there are &quot;ongoing discussions&quot; about a Rocky prequel television series, which he hopes will land on a streaming service and the series will likely follow a young Rocky Balboa as a professional boxing hopeful. Stallone said producer Irwin Winkler is hesitant on making the series saying that &quot;There was some conflict there, yes. He felt in his mind that &quot;Rocky&quot; was primarily a feature film, and he didn't see it as being translated for cable, so there was a big bone of contention.&quot;&lt;ref&gt;{{cite web |url=https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/ |title=Sylvester Stallone Feels Robbed of an Ownership Stake in 'Rocky': 'I Was Furious' |work=[[Variety (magazine)|Variety]] |date=July 23, 2019 |access-date=July 23, 2019 |archive-date=October 31, 2021 |archive-url=https://web.archive.org/web/20211031124140/https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/ |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://popculture.com/movies/2019/07/23/sylvester-stallone-reveals-rocky-sequel-prequel-in-development/ |title=Sylvester Stallone Reveals 'Rocky' Sequel and Prequel Are in Development |work=popculture.com |date=July 16, 2019 |access-date=July 23, 2019 |archive-date=July 24, 2019 |archive-url=https://web.archive.org/web/20190724165447/https://popculture.com/movies/2019/07/23/sylvester-stallone-reveals-rocky-sequel-prequel-in-development/ |url-status=live }}&lt;/ref&gt; That year, Stallone hand-picked [[Derek Wayne Johnson]] to direct and produce a documentary on the making of the original ''[[Rocky]]'', entitled ''[[40 Years of Rocky]]'' (2020). The documentary features Stallone narrating behind-the-scenes footage from the making of the film.&lt;ref name=&quot;:1&quot;&gt;Drown, Michelle. [http://www.independent.com/news/2017/jan/26/john-avildsen-king-underdogs/ &quot;John G. Avildsen: King of the Underdogs Director Derek Wayne Johnson&quot;] {{Webarchive|url=https://web.archive.org/web/20180718030602/http://www.independent.com/news/2017/jan/26/john-avildsen-king-underdogs/ |date=July 18, 2018 }}, ''The Santa Barbara Independent'', Santa Barbara, CA, January 26, 2017. Retrieved on February 16, 2017.&lt;/ref&gt;

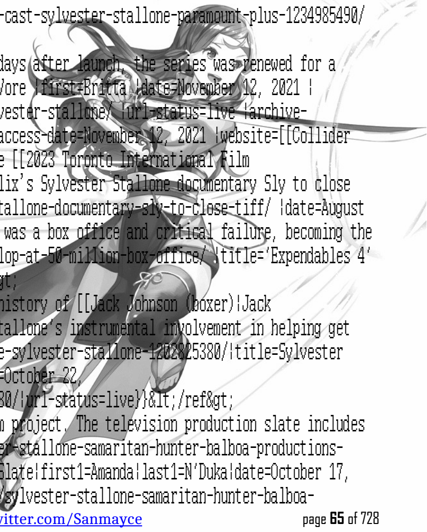
137 Stallone formed a film studio named Balboa Productions with Braden Aftergood in March 2018, where Stallone will serve as co-producer for each of their projects. The studio signed a multi-year collaboration deal with Starlight Culture Entertainment to develop projects for film and television.&lt;ref&gt;{{Cite web|url=https://www.hollywoodreporter.com/news/sylvester-stallone-launches-balboa-productions-1115792|title=Sylvester Stallone Launches Balboa Productions|website=The Hollywood Reporter|date=May 30, 2018|access-date=October 22, 2018|archive-date=September 27, 2018|archive-url=https://web.archive.org/web/20180927124952/https://www.hollywoodreporter.com/news/sylvester-stallone-launches-balboa-productions-1115792|url-status=live}}&lt;/ref&gt; In May 2018, a fifth installment in the [[Rambo (franchise)|''Rambo'' franchise]] was announced, and in August 2018, [[Adrian Grünberg]] was confirmed as the director.&lt;ref&gt;{{cite web|url=https://screenrant.com/rambo-5-director/|title=Rambo 5 Reportedly Recruits Get the Gringo Director|first=David|last=Stephens|work=Screen Rant|date=August 18, 2018|access-date=October 2, 2018|archive-date=August 19, 2018|archive-url=https://web.archive.org/web/20180819050055/https://screenrant.com/rambo-5-director/|url-status=live}}&lt;/ref&gt; ''[[Rambo: Last Blood]]'' began filming by September 2018, with a script co-written by Stallone, who also reprised his role as [[Vietnam War]] veteran John Rambo.&lt;ref&gt;{{cite web|last=Wiseman|first=Andrew|work=[[Deadline Hollywood|Deadline]]|url=https://deadline.com/2018/05/sylvester-stallone-in-line-to-take-on-mexican-cartel-in-rambo-5-1202383468/|title=Sylvester Stallone In Line To Take on Mexican Cartel in Rambo 5|date=May 5, 2018|access-date=May 7, 2018|archive-date=February 14, 2019|archive-url=https://web.archive.org/web/20190214104758/https://deadline.com/2018/05/sylvester-stallone-in-line-to-take-on-mexican-cartel-in-rambo-5-1202383468/|url-status=live}}&lt;/ref&gt; The plot centers around Rambo infiltrating a [[Mexican Drug War|Mexican drug cartel]] to rescue a family friend's daughter.&lt;ref name=&quot;ScreenDaily&quot;&gt;{{cite magazine|last=Kay|first=Jeremy|url=https://www.screendaily.com/sylvester-stallone-lining-up-rambo-v-exclusive/5128882.article|title=Sylvester Stallone lining up 'Rambo V' (exclusive)|magazine=[[Screen Daily]]|date=May 5, 2018|access-date=May 8, 2018|archive-date=May 8, 2018|archive-url=https://web.archive.org/web/20180508002459/https://www.screendaily.com/sylvester-stallone-lining-up-rambo-v-exclusive/5128882.article|url-status=live}}&lt;/ref&gt; The film, which was released on September 20, 2019, in the United States,&lt;ref name=&quot;IGN&quot;&gt;{{cite news|last=Gilyadov|first=Alex|url=https://www.ign.com/articles/2018/05/08/rambo-5-reportedly-in-the-works-with-sylvester-stallone|title=Rambo 5 Confirmed by Stallone, Coming Fall 2019|website=IGN|date=May 8, 2018|access-date=May 8, 2018|archive-date=May 9, 2018|archive-url=https://web.archive.org/web/20180509082229/https://www.ign.com/articles/2018/05/08/rambo-5-reportedly-in-the-works-with-sylvester-stallone|url-status=live}}&lt;/ref&gt; grossed $18.9 million in its opening weekend, the best debut of the franchise.&lt;ref&gt;{{Cite web|url=https://comicbook.com/movies/2019/08/29/rambo-last-blood-projected-franchise-best-opening/|title=Rambo: Last Blood Projected For Franchise Best Opening|access-date=August 31, 2019|url-status=dead|archive-date=August 31, 2019|archive-url=https://web.archive.org/web/20190831005659/https://comicbook.com/movies/2019/08/29/rambo-last-blood-projected-franchise-best-opening/}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |url=https://deadline.com/2019/08/rambo-last-blood-ad-astra-downton-abbey-box-office-projections-1202707434/ |title=Rambo: Last Blood' Eyes Franchise Best $24M+ Opening, Will Toss 'Ad Astra' Out Of Orbit &amp; Crash 'Downton Abbey's Party |last=D'Alessandro |first=Anthony |website=[[Deadline Hollywood]] |date=August 29, 2019 |access-date=August 29, 2019 |archive-date=August 29, 2019 |archive-url=https://web.archive.org/web/20190829181842/https://deadline.com/2019/08/rambo-last-blood-ad-astra-downton-abbey-box-office-projections-1202707434/ |url-status=live }}&lt;/ref&gt; The film grossed $91 million worldwide against a production budget of $50 million.

138 In late 2020, Stallone lent his voice as Rambo to the fighting video game ''[[Mortal Kombat 11]]'', as part of the game's 2nd Kombat Pack.&lt;ref&gt;{{Cite web|last=Amore|first=Samson|date=October 8, 2020|title=Sylvester Stallone to Voice Rambo in 'Mortal Kombat 11 Ultimate'|url=https://www.thewrap.com/rambo-mortal-kombat-11-ultimate-sylvester-stallone/|access-date=July 5, 2021|website=TheWrap|archive-date=June 23, 2021|archive-url=https://web.archive.org/web/20210623065435/https://www.thewrap.com/rambo-mortal-kombat-11-ultimate-sylvester-stallone/|url-status=live}}&lt;/ref&gt; In 2021, he voiced [[King Shark]] in the [[DC Extended Universe]] film ''[[The Suicide Squad (film)|The Suicide Squad]]''.&lt;ref&gt;{{cite web|url=https://www.irishtimes.com/culture/film/king-shark-sylvester-stallone-i-m-looking-for-love-it-s-the-loneliest-apex-predator-1.4624720|title=King Shark Sylvester Stallone: 'I'm looking for love. It's the loneliest apex predator'|publisher=The Irish Times|accessdate=August 10, 2021|archive-date=July 25, 2021|archive-url=https://web.archive.org/web/20210725231222/https://www.irishtimes.com/culture/film/king-shark-sylvester-stallone-i-m-looking-for-love-it-s-the-loneliest-apex-predator-1.4624720|url-status=live}}&lt;/ref&gt; In 2022, Stallone starred in ''[[Samaritan (film)|Samaritan]]'', a dark interpretation of the [[superhero]] genre, from a script written by Bragi Schut.&lt;ref&gt;{{cite web|first=Dave|last=McNary|url=https://variety.com/2019/film/news/sylvester-stallone-samaritan-mgm-1203145083/|title=Sylvester Stallone's Superhero Drama 'Samaritan' Lands at MGM|date=February 21, 2019|website=[[Variety (magazine)|Variety]]|access-date=December 8, 2019|archive-date=February 22, 2019|archive-url=https://web.archive.org/web/20190222012147/https://variety.com/2019/film/news/sylvester-stallone-samaritan-mgm-1203145083/|url-status=live}}&lt;/ref&gt; In November 2022, Stallone made his [[streaming television]] debut in ''[[Tulsa King]]'', in which he plays a Mafia boss. The 9-episode series was created by [[Taylor Sheridan]] and [[Terence Winter]] and premiered on [[Paramount+]].&lt;ref name=&quot;Mar22-cast&quot;&gt;{{cite web |last1=Andreeva |first1=Nellie |title=''Tulsa King'': Max Casella, Domenick Lombardozzi, Vincent Piazza &amp; Jay Will Join Sylvester Stallone In Paramount+ Series |url=https://deadline.com/2022/03/tulsa-king-max-casella-domenick-lombardozzi-vincent-piazza-jay-cast-sylvester-stallone-paramount-plus-1234985490/ |website=[[Deadline Hollywood]] |publisher=[[Penske Media Corporation]] |date=March 24, 2022 |access-date=May 9, 2022 |archive-date=April 15, 2022 |archive-url=https://web.archive.org/web/20220415095241/https://deadline.com/2022/03/tulsa-king-max-casella-domenick-lombardozzi-vincent-piazza-jay-cast-sylvester-stallone-paramount-plus-1234985490/ |url-status=live }}&lt;/ref&gt; A second season is in development.

139 In May 2023 Stallone and his immediate family (wife and three daughters) starred in a reality television series ''[[The Family Stallone]]''. Seven days after launch, the series was renewed for a second season. That same month, Stallone reprised his role as Stakar Ogord in ''[[Guardians of the Galaxy Vol. 3]]''.&lt;ref&gt;{{Cite web |last=DeVore |first=Britta |date=November 12, 2021 |title=Sylvester Stallone Confirms His Return to the MCU in 'Guardians of the Galaxy Vol. 3' |url=https://collider.com/guardians-of-the-galaxy-3-sylvester-stallone/ |url-status=live |archive-url=https://web.archive.org/web/20211113035650/https://collider.com/guardians-of-the-galaxy-3-sylvester-stallone/ |archive-date=November 13, 2021 |access-date=November 12, 2021 |website=[[Collider (website)|Collider]]}}&lt;/ref&gt; ''[[Sly (film)|Sly]]'', a documentary film about Stallone by [[Thom Zimny]], premiered as the closing film of the [[2023 Toronto International Film Festival]].&lt;ref&gt;[https://www.theglobeandmail.com/arts/film/tiff/article-netflixs-sylvester-stallone-documentary-sly-to-close-tiff/ &quot;Netflix's Sylvester Stallone documentary Sly to close TIFF&quot;] {{Webarchive|url=https://web.archive.org/web/20230810225254/https://www.theglobeandmail.com/arts/film/tiff/article-netflixs-sylvester-stallone-documentary-sly-to-close-tiff/ |date=August 10, 2023 }}. ''[[The Globe and Mail]]'', August 8, 2023.&lt;/ref&gt; In October 2023, he reprised role as Barney Ross in ''[[Expend4bles]]'', which was a box office and critical failure, becoming the lowest-grossing film in the franchise.&lt;ref&gt;{{cite web |url=https://www.forbes.com/sites/markhughes/2023/09/25/expendables-4-is-another-2023-flop-at-50-million-box-office/ |title='Expendables 4' Is Another 2023 Flop At $50 Million Box Office |work=Forbes |last=Hughes |first=Mark |date=September 25, 2023 |access-date=April 3, 2024}}&lt;/ref&gt;

140 Following the releases of ''Creed II'' and ''Rambo V: Last Blood'', Balboa Productions has had an extensive production slate. A film depicting the history of [[Jack Johnson (boxer)|Jack &quot;Galveston Giant&quot; Johnson]], the first African-American boxing heavyweight champion, is in development. The project was announced after Stallone's instrumental involvement in helping get Johnson a posthumous pardon from US President [[Donald Trump]].&lt;ref&gt;{{Cite web|url=https://variety.com/2018/film/news/jack-johnson-boxer-movie-sylvester-stallone-1202825380/|title=Sylvester Stallone to Make Movie About Trump-Pardoned Boxer Jack Johnson|first1=Dave|last1=McNary|date=May 30, 2018|access-date=October 22, 2018|archive-date=October 22, 2018|archive-url=https://web.archive.org/web/20181022073457/https://variety.com/2018/film/news/jack-johnson-boxer-movie-sylvester-stallone-1202825380/|url-status=live}}&lt;/ref&gt;

141 Stallone wrote ''[[Levon's Trade]]'' from a book series created by [[Chuck Dixon]], startng as a television adaptation and transitioning into a film project. The television production slate includes a series adaptation of Charles Sailor's ''Second Son'' being written by Rob Williams.&lt;ref&gt;{{Cite web|url=https://deadline.com/2018/10/sylvester-stallone-samaritan-hunter-balboa-productions-ghost-book-deal-1202483446/|title=Sylvester Stallone To Star In 'Samaritan' &amp; 'Hunter', Acquires FBI Memoir As His Balboa Productions Ramps Up Slate|first1=Amanda|last1=N'Duka|date=October 17, 2018|access-date=October 22, 2018|archive-date=October 22, 2018|archive-url=https://web.archive.org/web/20181022073513/https://deadline.com/2018/10/sylvester-stallone-samaritan-hunter-balboa-



productions-ghost-book-deal-1202483446/|url-status=live}}&lt;/ref&gt;

142 Stallone announced in early May 2020 that a sequel to 1993's ''[[Demolition Man (film)|Demolition Man]]'' is in the works: &quot;I think it's coming. We're working on it right now with Warner Brothers. It's looking fantastic. So, that should come out, that's going to happen&quot;.&lt;ref&gt;{{Cite web|date=May 5, 2020|title=Demolition Man 2 in the Works at Warner, Says Sylvester Stallone|url=https://www.denofgeek.com/movies/demolition-man-2-sylvester-stallone-warner/|access-date=June 30, 2020|website=Den of Geek|archive-date=May 12, 2020|archive-url=https://web.archive.org/web/20200512185456/https://www.denofgeek.com/movies/demolition-man-2-sylvester-stallone-warner/|url-status=live}}&lt;/ref&gt;

143 Asked in February 2008 which of the icons (Rocky or Rambo) he would rather be remembered for, Stallone said &quot;it's a tough one, but ''Rocky'' is my first baby, so Rocky.&quot;&lt;ref&gt;{{Cite web|title=Sylvester Stallone: Rambo Returns, video interview with STV|url=http://www.stv.tv/content/out/film/videointerviews/display.html?id=opencms:/out/films/video_interviews/sylvester_stallone_rambo_video_interview|url-status=dead|archive-url=https://web.archive.org/web/20080518053857/http://www.stv.tv/content/out/film/videointerviews/display.html?id=opencms%3A%2Fout%2Ffilms%2Fvideo_interviews%2Fsylvester_stallone_rambo_video_interview|archive-date=May 18, 2008}}&lt;/ref&gt; He also stated that Rocky could be interpreted as the &quot;[[conscious]]&quot; and Rambo as the &quot;[[Unconscious mind|unconscious]]&quot; of the same character.&lt;ref&gt;Le Cercle, 2008-02-09, cinema talk-show on French television channel [[Canal+ (French TV channel)|Canal+]], during a review of ''Rambo'', cited by Marie Sauvion: &quot;Et quand il [Stallone] se mêle de psychologie, voire de psychanalyse, alors il a un discours, il dit, Rocky c'est la conscience, en gros, d'un même personnage, et Rambo ce serait son inconscient.&quot; / &quot;And when he speaks in terms of psychology, or even psychoanalysis, he's got a discourse, he says, Rocky is the conscious, in broad strokes, of the same character, and Rambo would be his unconscious.&quot;&lt;/ref&gt;

144 [[File:Sylvester Stallone Brigitte Nielsen (cropped).jpg|thumb|upright=1|Stallone with then-wife [[Brigitte Nielsen]], President [[Ronald Reagan]], and First Lady [[Nancy Reagan]] at the [[White House]] in 1985]]

145 Stallone married model and actress [[Brigitte Nielsen]] on December 15, 1985, in [[Beverly Hills, California]]. Their marriage (which lasted two years) and their subsequent divorce were highly publicized by the tabloid press.&lt;ref&gt;Susan Zannos, ''Male Fitness Stars of TV and the Movies: Featuring Profiles of Sylvester Stallone, John Travolta, Bruce Willis, and Wesley Snipes'', Mitchell Lane Publishers, 2000, page 27&lt;/ref&gt;&lt;ref&gt;{{cite news |url=https://news.google.com/newspapers?nid=1755&amp;dat=19870723&amp;id=vqMcAAAAIBAJ&amp;sjid=RXAEAAAAIBAJ&amp;pg=7140,3399728 |last=Broeske |first=Pat H. |title=Stallone Divorce Stops Tabloid Presses |date=July 23, 1987 |newspaper=[[Sarasota Herald-Tribune]] |agency=[[Los Angeles Times-Washington Post News Service]] |location=Hollywood |via=Google News |access-date=November 20, 2015 |archive-date=June 26, 2019 |archive-url=https://web.archive.org/web/20190626193959/https://news.google.com/newspapers?nid=1755&amp;dat=19870723&amp;id=vqMcAAAAIBAJ&amp;sjid=RXAEAAAAIBAJ&amp;pg=7140,3399728 |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite news |url=https://www.nytimes.com/1997/08/10/movies/stallone-seeks-a-serious-turn-for-the-better.html |last=Gabriel |first=Trip |title=Stallone Seeks a Serious Turn for the Better |date=August 10, 1997 |newspaper=[[The New York Times]] |access-date=February 18, 2017 |archive-date=March 29, 2012 |archive-url=https://web.archive.org/web/20120329085540/http://www.nytimes.com/1997/08/10/movies/stallone-seeks-a-serious-turn-for-the-better.html |url-status=live }}&lt;/ref&gt;

146 In May 1997, Stallone and Jennifer Flavin married. The couple has three daughters, Sophia, [[Sistine Stallone|Sistine]], and Scarlet.&lt;ref name=&quot;PeopleDaughters&quot;&gt;{{cite news|url=http://people.com/movies/sylvester-stallones-daughters-what-to-know-about-sophia-sistine-scarlet/|title=Sylvester Stallone's Daughters: What to Know About Miss Golden Globes Sophia, Sistine &amp; Scarlet|author=Michelle Miller|newspaper=People|date=January 5, 2017|access-date=April 23, 2017|archive-date=September 13, 2023|archive-url=https://web.archive.org/web/20230913034559/https://people.com/movies/sylvester-stallones-daughters-what-to-know-about-sophia-sistine-scarlet/|url-status=live}}&lt;/ref&gt; His daughters were chosen to share the role of [[Golden Globe Ambassador]] at the [[74th Golden Globe Awards]].&lt;ref&gt;{{cite news|title=Who is this year's Miss Golden Globe? All three of Sylvester Stallone's daughters|url=https://www.latimes.com/entertainment/la-et-golden-globes-2017-live-sylvester-stallone-s-girls-are-all-miss-1483861175-htmlstory.html|newspaper=[[Los Angeles Times]]|date=January 8, 2017|last=D'Zurilla|first=Christie|access-date=February 19, 2020|archive-date=April 26, 2019|archive-url=https://web.archive.org/web/20190426143634/https://www.latimes.com/entertainment/la-et-golden-globes-2017-live-sylvester-stallone-s-girls-are-all-miss-1483861175-htmlstory.html|url-status=live}}&lt;/ref&gt; On August 19, 2022, [[Palm Beach County, Florida|Palm Beach County]] records showed that after 25 years of marriage, Jennifer Flavin had filed for &quot;[[Divorce|dissolution of marriage]] and other relief&quot;.&lt;ref&gt;{{cite news |last=Marr |first=Madeleine |date=August 24, 2022 |title=They were married 25 years. Now Sly Stallone's wife has filed for divorce in Florida |work=[[Miami Herald]] |url=https://www.miamiherald.com/miami-com/miami-com-news/article264837969.html |access-date=August 26, 2022 |archive-date=August 24, 2022 |archive-url=https://web.archive.org/web/20220824185544/https://www.miamiherald.com/miami-com/miami-com-news/article264837969.html |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite news |last=Whigham |first=Julius |date=August 25, 2022 |title=Sylvester Stallone divorce: Jennifer Flavin files in county, cites 'waste of marital assets' |work=[[Palm Beach Post]] |url=https://www.palmbeachpost.com/story/news/local/2022/08/25/jennifer-flavin-seeks-divorce-rocky-rambo-star-sylvester-stallone/7887378001/ |access-date=August 26, 2022 |archive-date=August 25, 2022 |archive-url=https://web.archive.org/web/20220825220351/https://www.palmbeachpost.com/story/news/local/2022/08/25/jennifer-flavin-seeks-divorce-rocky-rambo-star-sylvester-stallone/7887378001/ |url-status=live }}&lt;/ref&gt; On September 23, 2022, the couple reconciled.&lt;ref&gt;{{cite web |url=https://people.com/movies/sylvester-stallone-wife-jennifer-flavin-reconcile-after-divorce-filing/ |title=Sylvester Stallone and Wife Jennifer Flavin Reconcile 1 Month After She Filed for Divorce |last1=VanHoose |first1=Benjamin |last2=Leonard |first2=Elizabeth |date=September 23, 2022 |website=People |access-date=July 5, 2023 |archive-date=June 25, 2023 |archive-url=https://web.archive.org/web/20230625175219/https://people.com/movies/sylvester-stallone-wife-jennifer-flavin-reconcile-after-divorce-filing/ |url-status=live }}&lt;/ref&gt;

147 Stallone's 48-year-old half-sister, Toni Ann Filiti, died of lung cancer on August 26, 2012.&lt;ref&gt;{{cite news |last=Dillon |first=Nancy |url=http://www.nydailynews.com/news/national/tragedy-strikes-sylvester-stallone-family-expendables-star-loses-half-sister-cancer-article-1.1145688 |title=Sylvester Stallone's half-sister Toni Ann Filiti dies of cancer at 48 |date=August 27, 2012 |work=[[Daily News (New York)|Daily News]] |access-date=June 19, 2013 |archive-date=December 28, 2013 |archive-url=https://web.archive.org/web/20131228102440/http://www.nydailynews.com/news/national/tragedy-strikes-sylvester-stallone-family-expendables-star-loses-half-sister-cancer-article-1.1145688 |url-status=live }}&lt;/ref&gt;

148 Known for physically demanding roles and his willingness to do the majority of his own stunts, Stallone has suffered various injuries during his acting career. During the filming of ''[[Escape to Victory]]'', he broke a finger trying to save a [[Penalty kick (association football)|penalty kick]] against [[Pelé]].&lt;ref&gt;{{cite news |url=https://www.bbc.co.uk/sport/0/football/25666378 |title=Premier League predictions: Lawro v Robert De Niro &amp; Sly Stallone |work=BBC Sport |date=January 12, 2014 |access-date=May 23, 2014 |archive-date=January 20, 2014 |archive-url=https://web.archive.org/web/20140120001837/http://www.bbc.co.uk/sport/0/football/25666378 |url-status=live }}&lt;/ref&gt; For a scene in ''Rocky IV'', he told [[Dolph Lundgren]], &quot;Punch me as hard as you can in the chest.&quot; He later said, &quot;Next thing I know, I was in intensive care at [[Saint John's Health Center|St. John's Hospital]] for four days. It's stupid!&quot;&lt;ref&gt;The Expendables DVD: Comic-Con 2010 Panel&lt;/ref&gt;&lt;ref&gt;{{cite web | title=Stallone: Lundgren Put Me in Hospital During 'Rocky IV' | website=Muscle &amp; Fitness | date=August 11, 2014 | url=https://www.muscleandfitness.com/athletes-celebrities/news/sylvester-stallone-lundgren-put-me-hospital-during-rocky-iv/ | access-date=October 11, 2020 | archive-date=November 8, 2020 | archive-url=https://web.archive.org/web/20201108133537/https://www.muscleandfitness.com/athletes-celebrities/news/sylvester-stallone-lundgren-put-me-hospital-during-rocky-iv/ | url-status=live }}&lt;/ref&gt; While filming a fight scene with [[Stone Cold Steve Austin|&quot;Stone Cold&quot; Steve Austin]] for ''[[The Expendables (2010 film)|The Expendables]]'', he broke his neck and required the insertion of a metal plate.&lt;ref&gt;{{cite news |url=http://news.bbc.co.uk/1/hi/8443066.stm |title=Sylvester Stallone injures neck in fight scenes |work=BBC News |date=January 6, 2010 |access-date=September 4, 2010 |archive-date=April 3, 2022 |archive-url=https://web.archive.org/web/20220403232027/http://news.bbc.co.uk/1/hi/8443066.stm |url-status=live }}&lt;/ref&gt;

149 In February 2001, an exotic dancer named Margie Carr filed a lawsuit against Stallone, accusing him of rape while at a Santa Monica fitness center where they both worked out on February 26, 2000. A lawyer for Stallone denied the claim, saying she sold the story to ''[[Globe (tabloid)|Globe]]'' the month before the lawsuit.&lt;ref&gt;{{Cite news|date=March 8, 2001|title=Stallone sued for 'sexual assault'|url=http://news.bbc.co.uk/2/hi/entertainment/1209746.stm|access-date=October 11, 2020|archive-date=October 17, 2020|archive-url=https://web.archive.org/web/20201017012900/http://news.bbc.co.uk/2/hi/entertainment/1209746.stm|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;mercurynews.com&quot;&gt;{{Cite web|date=November 17, 2017|title=Sylvester Stallone has a history of denying women's claims of sexual abuse|url=https://www.mercurynews.com/2017/11/17/sylvester-stallone-has-a-history-of-denying-womens-claims-of-sexual-abuse/|access-date=October 11, 2020|website=The Mercury News|archive-date=October 11, 2020|archive-url=https://web.archive.org/web/20201011032650/https://www.mercurynews.com/2017/11/17/sylvester-stallone-has-a-history-of-denying-womens-claims-of-sexual-abuse/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|author=&lt;!--Staff writer(s); no by-line.--&gt;|title=Stripper Sues Stallone for Assault|url=https://abcnews.go.com/Entertainment/story?id=108683&amp;page=1|access-date=October 11, 2020|website=ABC News|archive-date=August 8, 2022|archive-url=https://web.archive.org/web/20220808045119/https://abcnews.go.com/Entertainment/story?id=108683&amp;page=1|url-status=live}}&lt;/ref&gt;

150 In 2007, customs officials in Australia discovered 48 vials of the synthetic human growth hormone [[Jintropin]] in Stallone's luggage.&lt;ref&gt;{{cite web |last=Childs |first=Dan |title=Will Stallone's HGH Secret Start a Trend? |url=https://abcnews.go.com/Health/ActiveAging/story?id=3176015 |publisher=ABC News |access-date=June 29, 2020 |archive-date=February 1, 2008 |archive-url=https://web.archive.org/web/20080201233637/https://abcnews.go.com/Health/ActiveAging/story?id=3176015 |url-status=live }}&lt;/ref&gt; In a court hearing on May 15, he pleaded guilty to two charges of possessing a controlled substance.&lt;ref&gt;{{cite web |date=May 15, 2007 |title=Sylvester Stallone Pleads Guilty to Bringing Human Growth Hormone Into Australia |url=https://www.foxnews.com/story/sylvester-stallone-pleads-guilty-to-bringing-human-growth-hormone-into-australia |access-date=April 3, 2019 |work=Fox News |archive-date=April 3, 2019 |archive-url=https://web.archive.org/web/20190403183008/https://www.foxnews.com/story/sylvester-stallone-pleads-guilty-to-bringing-human-growth-hormone-into-australia |url-status=live }}&lt;/ref&gt;

151 In 2016, a report from the Las Vegas Metropolitan Police Department was published stating that Stallone was accused of sexual assault by a 16-year-old girl while he was shooting a film in [[Las Vegas]] in 1986. The teen reportedly said that Stallone, then 40, forced her into a [[threesome]] with his bodyguard.&lt;ref&gt;{{cite news|first=Doug|last=Poppa|url=http://baltimorepostexaminer.com/ask-sylvester-stallone-40-year-old-man-group-sex-masturbate-front-minor/2017/11/10|title=Ask Sylvester Stallone if a 40-year-old man should have group sex with and masturbate in front of a minor|newspaper=Baltimore Post-Examiner|date=November 10, 2017|access-date=June 15, 2018|archive-date=August 24, 2019|archive-url=https://web.archive.org/web/20190824132512/https://baltimorepostexaminer.com/ask-sylvester-stallone-40-year-old-man-group-sex-masturbate-front-minor/2017/11/10|url-status=live}}&lt;/ref&gt; A representative for Stallone denied the allegation.&lt;ref&gt;{{cite web|first=Andrea|last=Mandell|url=https://www.usatoday.com/story/life/people/2017/11/16/rep-sexual-assault-claim-against-sylvester-stallone-categorically-false/872915001/|title=Rep: Sexual assault claim against Sylvester Stallone 'categorically false'|newspaper=[[USA Today]]|date=November 17, 2017|access-date=June 15, 2018|archive-date=August 24, 2019|archive-url=https://web.archive.org/web/20190824132456/https://www.usatoday.com/story/life/people/2017/11/16/rep-sexual-assault-claim-against-sylvester-stallone-categorically-false/872915001/|url-status=live}}&lt;/ref&gt; Stallone's ex-wife, [[Brigitte Nielsen]], later came to his defense, saying that she was with him at the time of the alleged assault. Stallone's ''Over the Top'' costar [[David Mendenhall]] also defended Stallone, denying claims that he introduced Stallone to the girl in question.&lt;ref&gt;{{Cite web|url=https://www.tmz.com/2017/11/20/sylvester-stallone-brigitte-nielsen-sexual-assault-rape/|title=Sylvester Stallone's Ex-Wife Brigitte Nielsen Says Sexual Assault is a Lie|website=[[TMZ]]|date=November 20, 2017|access-date=February 19, 2020|archive-date=August 7, 2020|archive-url=https://web.archive.org/web/20200807213644/https://www.tmz.com/2017/11/20/sylvester-stallone-brigitte-nielsen-sexual-assault-rape/|url-status=live}}&lt;/ref&gt;

152 In November 2017, a woman accused Stallone of sexually assaulting her at his [[Santa Monica, California|Santa Monica]] office in the early 1990s. Stallone denied the claim.&lt;ref&gt;{{cite news|url=https://www.bbc.co.uk/news/entertainment-arts-42453515|title=Sylvester Stallone denies rape as police investigate|work=[[BBC News]]|date=December 22, 2017|access-date=July 21, 2018|archive-date=August 28, 2019|archive-url=https://web.archive.org/web/20190828135812/https://www.bbc.co.uk/news/entertainment-arts-42453515|url-status=live}}&lt;/ref&gt; His attorney revealed the accuser filed a report after an entertainment website declined to pick up the story.&lt;ref&gt;{{cite news|url=https://www.latimes.com/local/lanow/la-me-stallone-sex-assault-da-20180613-story.html|

```
title=Sexual assault allegation against actor Sylvester Stallone|newspaper=[[Los Angeles Times]]|date=June 13, 2018|access-date=February 19, 2020|archive-date=November 14, 2019|archive-url=https://web.archive.org/web/20191114142427/https://www.latimes.com/local/lanow/la-me-stallone-sex-assault-da-20180613-story.html|url-status=live}}&lt;/ref&gt; Stallone's attorneys also stated that while the actor had a consensual relationship with the accuser in 1987, they had two witnesses who refuted the claims.&lt;ref&gt;{{cite news|url=https://news.avclub.com/the-los-angeles-da-is-investigating-sexual-assault-alle-1826833725|title=The Los Angeles DA is investigating sexual assault allegations against Sylvester Stallone|newspaper=[[Los Angeles Times]]|date=June 14, 2018|access-date=October 3, 2018|archive-date=October 3, 2018|archive-url=https://web.archive.org/web/20181003101005/https://news.avclub.com/the-los-angeles-da-is-investigating-sexual-assault-alle-1826833725|url-status=live}}&lt;/ref&gt; In June 2018, the Los Angeles District Attorney's office confirmed an investigation, stating that the Santa Monica Police Department had presented a [[sex crimes]] case against Stallone to a special prosecution task force for review.&lt;ref&gt;{{cite news|url=https://www.usatoday.com/story/life/2018/06/14/sylvester-stallone-sex-crime-allegation-under-review-prosecutors/701542002/|title=Sylvester Stallone sex-crime allegation under review by Los Angeles prosecutors|newspaper=[[USA Today]]|date=June 14, 2018|access-date=June 15, 2018|archive-date=August 24, 2019|archive-url=https://web.archive.org/web/20190824133959/https://www.usatoday.com/story/life/2018/06/14/sylvester-stallone-sex-crime-allegation-under-review-prosecutors/701542002/|url-status=live}}&lt;/ref&gt; In October 2018, the Los Angeles District Attorney's office made the decision not to charge Stallone for the alleged assault, as no witnesses corroborated the allegations. Stallone in turn filed a police report regarding her lying on an official document.&lt;ref&gt;{{cite web|language=it|url=https://www.chedonna.it/2018/10/31/accusato-di-violenza-sessuale-sylvester-stallone-non-sara-processato/|title=Accusato di violenza sessuale, Sylvester Stallone non sarà processato|website=Chedonna|date=October 31, 2018|access-date=October 31, 2018|archive-date=August 24, 2019|archive-url=https://web.archive.org/web/20190824132458/https://www.chedonna.it/2018/10/31/accusato-di-violenza-sessuale-sylvester-stallone-non-sara-processato/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.tmz.com/2018/10/30/sylvester-stallone-no-charges-sexual-assault-rape-accuser/|title=Sylvester Stallone - D.A. Rejects 1990 Rape Case|date=October 30, 2018 |publisher=TMZ|access-date=October 31, 2018|archive-date=October 30, 2018|archive-url=https://web.archive.org/web/20181030223156/http://www.tmz.com/2018/10/30/sylvester-stallone-no-charges-sexual-assault-rape-accuser/|url-status=live}}&lt;/ref&gt;
Stallone was baptized and raised a devout [[Catholic Church|Catholic]], but stopped going to church as his acting career progressed. He rediscovered his childhood faith when his daughter was born ill in 1996, and he again became a strict Catholic by late 2006.&lt;ref&gt;{{cite web|work=Catholic Online |url=http://www.catholic.org/national/national_story.php?id=22474 |title='Rocky' Stallone back in church as new movie in theaters |publisher=Catholic.org |access-date=September 4, 2010 |url-status=dead |archive-url=https://web.archive.org/web/20100820014415/http://www.catholic.org/national/national_story.php?id=22474 |archive-date=August 20, 2010 }}&lt;/ref&gt; The same year, he was interviewed by [[Pat Robertson]] from the [[Christian Broadcasting Network]]'s ''[[The 700 Club|700 Club]]''. He stated that he spent much of his previous time in Hollywood &quot;losing his way&quot; because temptation abounded, but later put things &quot;in God's hands&quot;.&lt;ref&gt;[http://www.cbn.com/tv/1365948765001 Sylvester Stallone On Faith, Integrity, And Rocky] {{Webarchive|url=https://web.archive.org/web/20190824132511/https://www.cbn.com/tv/1365948765001 |date=August 24, 2019 }}. CBNTV.&lt;/ref&gt; However, he told ''[[GQ]]'' magazine in 2010, &quot;I'm pretty spiritual; I believe a lot in the spirit of man. I'm certainly not an atheist... I was baptized Catholic, but I don't belong to a structured church. I have no opposition to it. I think there's great nuggets of knowledge in there, some wonderful rules to live by. Then the flip side is the amount of agony that's caused.&quot;&lt;ref name=GQ&gt;{{cite news|url=https://www.gq.com/entertainment/celebrities/201009/sylvester-stallone-yo-michael-hainey-cop-land-rocky-rambo?currentPage=4|title=Yo.|work=[[GQ]]|date=September 2010|access-date=December 30, 2010|first=Michael|last=Hainey|quote=Are you a religious man? I'm pretty spiritual; I believe a lot in the spirit of man. I'm certainly not an atheist... I was baptized Catholic, but I don't belong to a structured church. I have no opposition to it. I think there's great nuggets of knowledge in there, some wonderful rules to live by. Then the flip side is the amount of agony that's caused, which is, excuse me?|archive-date=July 11, 2011|archive-url=https://web.archive.org/web/20110711120813/http://www.gq.com/entertainment/celebrities/201009/sylvester-stallone-yo-michael-hainey-cop-land-rocky-rambo?currentPage=4|url-status=live}}&lt;/ref&gt;
Stallone has supported several [[Republican Party (United States)|Republican]] politicians, but does not identify as a member of the Republican Party.&lt;ref name=GOP&gt;{{cite web|url=https://variety.com/2016/film/news/sylvester-stallone-donald-trump-1201675595/|title=Sylvester Stallone on Donald Trump, Republicans and Running for Office|first=Ramin|last=Setoodeh|newspaper=[[Variety (magazine)|Variety]]|location=Los Angeles|date=January 8, 2016|access-date=May 9, 2017|archive-date=September 6, 2019|archive-url=https://web.archive.org/web/20190906232813/https://variety.com/2016/film/news/sylvester-stallone-donald-trump-1201675595/|url-status=live}}&lt;/ref&gt; He has also donated to the [[Democratic National Committee]] and to [[Democratic Party (United States)|Democrats]] such as [[Joe Biden]] and [[Chris Dodd]]. He is a staunch advocate of [[gun control]] despite his otherwise [[Conservatism in the United States|conservative]] views, and has been described as &quot;the most anti-gun celebrity in Hollywood&quot;.&lt;ref&gt;{{cite news |first=Asawin |last=Suebsaeng |url=http://www.thedailybeast.com/articles/2014/08/14/rambo-hates-guns-how-sylvester-stallone-became-the-most-anti-gun-celeb-in-hollywood.html |title=Rambo Hates Guns: How Sylvester Stallone Became the Most Anti-Gun Celeb in Hollywood |newspaper=[[The Daily Beast]] |date=August 14, 2014 |access-date=July 7, 2016 |archive-date=May 4, 2017 |archive-url=https://web.archive.org/web/20170504095005/http://www.thedailybeast.com/articles/2014/08/14/rambo-hates-guns-how-sylvester-stallone-became-the-most-anti-gun-celeb-in-hollywood.html |url-status=live }}&lt;/ref&gt;
In 2015, upon the advice of a [[Vedas|Vedic]] scholar, Stallone performed a [[Hinduism|Hindu]] ''Tithi Shradh'' ritual (done for those who died by accident or murder) at [[Haridwar]] for his son Sage.&lt;ref&gt;{{Cite news|date=October 14, 2015|title=Sylvester Stallone has shradh performed in Haridwar for dead son|work=Hindustan Times|url=https://www.hindustantimes.com/hollywood/sylvester-stallone-has-shradh-performed-in-haridwar-for-dead-son/story-g5IE9JSLcADmosjwJiPWlO.html|access-date=August 23, 2021|archive-date=August 23, 2021|archive-url=https://web.archive.org/web/20210823083806/https://www.hindustantimes.com/hollywood/sylvester-stallone-has-shradh-performed-in-haridwar-for-dead-son/story-g5IE9JSLcADmosjwJiPWlO.html|url-status=live}}&lt;/ref&gt;
In the run-up to the [[2016 United States presidential election|2016 presidential election]], Stallone described [[Donald Trump]] as a &quot;[[Charles Dickens#Characters|Dickensian]] character&quot; and &quot;larger than life&quot; but did not endorse Trump or anyone else in the [[Republican Party presidential primaries, 2016|Republican primaries]].&lt;ref name=GOP /&gt; That same year, he declined an offer to become Chair of the [[National Endowment for the Arts]], citing a desire to work on issues related to veterans.&lt;ref&gt;{{cite news|url=https://www.vanityfair.com/hollywood/2016/12/sylvester-stallone-trump-arts|last=Desta|first=Yohana|title=Sylvester Stallone Isn't Interested in Trump's Offer After All|date=December 19, 2016|access-date=May 9, 2017|magazine=[[Vanity Fair (magazine)|Vanity Fair]]|archive-date=May 23, 2017|archive-url=https://web.archive.org/web/20170523153033/http://www.vanityfair.com/hollywood/2016/12/sylvester-stallone-trump-arts|url-status=live}}&lt;/ref&gt;
In 2023, Stallone donated $11,600 to [[Independent politician|independent]] Senator [[Kyrsten Sinema]]'s [[2024 United States Senate election in Arizona|re-election campaign in Arizona]].&lt;ref&gt;{{cite news|last=Kurtz|first=Judy|title=Celebs throw down cash in closely watched 2024 Senate races|url=https://thehill.com/blogs/in-the-know/4123124-celebs-throw-down-cash-in-closely-watched-2024-senate-races/|work=[[The Hill (newspaper)|the Hill]]|date=27 July 2023|access-date=July 27, 2023|archive-date=July 27, 2023|archive-url=https://web.archive.org/web/20230727223824/https://thehill.com/blogs/in-the-know/4123124-celebs-throw-down-cash-in-closely-watched-2024-senate-races/|url-status=live}}&lt;/ref&gt; Later that year, he and his family spent time with [[Pope Francis]] in a private meeting at the Vatican.&lt;ref&gt;{{cite web|url=https://deadline.com/2023/09/the-pope-surprises-sylvester-stallone-with-punch-vatican-meeting-1235541584/|title=The Pope surprises Sylvester Stallone with a punch in Vatican meeting|publisher=Deadline|date=September 9, 2023|access-date=September 11, 2023|archive-date=September 10, 2023|archive-url=https://web.archive.org/web/20230910151448/https://deadline.com/2023/09/the-pope-surprises-sylvester-stallone-with-punch-vatican-meeting-1235541584/|url-status=live}}&lt;/ref&gt;
{{main| Sylvester Stallone filmography| List of awards and nominations received by Sylvester Stallone}}
*Heart of Hollywood Award from the Board of Governors of the [[Cedars-Sinai Medical Center]] (2016)&lt;ref name=&quot;bhcouriercedarssinaiboardofgovernors&quot;&gt;{{cite news |date=October 10, 2016 |title=Cedars-Sinai Board Of Governors Gala To Honor Adele &amp; Beny Alagem and Sylvester Stallone |work=The Beverly Hills Courier}}&lt;/ref&gt;
{{Sylvester Stallone|state=expanded}}
|title = [[List of awards and nominations received by Sylvester Stallone|Awards for Sylvester Stallone]]
[[Category:Sylvester Stallone| ]]
On October 2, 2023, it was announced that CEO Bill Block would be leaving the company that week following the end of his contract with them, starting his own production company, BlockFilm, later that month.&lt;ref&gt;{{cite web |url= https://deadline.com/2023/10/miramax-ceo-bill-block-exits-1235561838/|title= Shocker! Shakeup At Miramax As CEO Bill Block Exits|date= October 2, 2023|access-date= October 2, 2023|first1= Mike Jr.|last1= Fleming|first2= Nellie|last2= Andreeva}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=Jason Statham To Star In David Ayer-Directed Sylvester Stallone Script 'Levon's Trade' From Black Bear &amp; Bill Block's BlockFilm – AFM Hot Package|website=[[Deadline Hollywood]]|first=Andreas|last=Wiseman|date=27 October 2023|access-date=28 October 2023|url=https://deadline.com/2023/10/jason-statham-david-ayer-sylvester-stallone-levons-trade-black-bear-bill-block-afm-1235585741/}}&lt;/ref&gt; In January 2024, [[Jonathan Glickman]] was reported to be in talks to join Miramax as CEO, with his company Panoramic Media expected to be acquired by Miramax if talks were successful.&lt;ref&gt;{{cite web|title=Jonathan Glickman in Talks to Lead Miramax as CEO|website=[[The Hollywood Reporter]]|first1=Kim|last1=Masters|first2=Aaron|last2=Couch|date=28 January 2024|access-date=7 March 2024|url=https://www.hollywoodreporter.com/movies/movie-news/jonathan-glickman-miramax-ceo-1235809126/}}&lt;/ref&gt; On April 2, 2024, Glickman was officially named Miramax's new CEO effective immediately. As part of the deal, Miramax also acquired certain film and TV projects from Glickman's company Panoramic Media.&lt;ref&gt;{{cite web|title=Miramax Names Jonathan Glickman CEO|website=[[Deadline Hollywood]]|first=Anthony|last=D'Alessandro|date=2 April 2024|access-date=2 April 2024|url=https://deadline.com/2024/04/miramax-jonathan-glickman-1235874073/}}&lt;/ref&gt;
Caine's other successful films (critically or financially) were the 1980 Golden Globe-nominated [[slasher film]] ''[[Dressed to Kill (1980 film)|Dressed to Kill]]'', the 1981 [[war film]] ''[[Escape to Victory]]'' featuring [[Sylvester Stallone]] and footballers from the 1960s and 1970s, including [[Pelé]] and [[Bobby Moore]], the 1982 film ''[[Deathtrap (film)|Deathtrap]]'', and ''[[Mona Lisa (film)|Mona Lisa]]'' (1986). In 1987, Caine narrated ''[[Hero (1987 film)|Hero]]'', the [[List of official FIFA World Cup films|official film]] of the [[1986 FIFA World Cup]].&lt;ref&gt;[http://www.fifafilms.com/fileadmin/fifafilms/user_upload/pdf/FIFAFilms_Factsheet_March_2012_very_low_res.pdf &quot;FIFA World Cup and Official FIFA Events: Programming&quot;] {{Webarchive|url=https://web.archive.org/web/20160417225504/http://www.fifafilms.com/fileadmin/fifafilms/user_upload/pdf/FIFAFilms_Factsheet_March_2012_very_low_res.pdf |date=17 April 2016 }}. FIFA Films. Retrieved 28 January 2013&lt;/ref&gt; That year he starred in the thriller and spy film ''[[The Fourth Protocol (film)|The Fourth Protocol]]'' alongside [[Pierce Brosnan]]. In 1988 he played Chief Insp. [[Frederick Abberline]] in the two-part TV drama ''[[Jack the Ripper (miniseries)|Jack the Ripper]]'', which co-starred [[Jane Seymour (actress)|Jane Seymour]] and was produced to coincide with the 100th anniversary of the [[Jack the Ripper]] murder spree in Victorian London.&lt;ref&gt;{{Cite web|url=https://www.radiotimesbacknumbers.com/index.php?route=product/product&amp;product_id=4195|title=TVT 1988/42 - 15-21 October 1988 (TVS and C4) JACK THE RIPPER (ITV) with cover photo of Michael Caine and Jane Seymour.|website=www.radiotimesbacknumbers.com|language=en|access-date=26 August 2021|archive-date=29 March 2019|archive-url=https://web.archive.org/web/20190329222607/https://www.radiotimesbacknumbers.com/index.php?route=product%2Fproduct&amp;product_id=4195|url-status=dead}}&lt;/ref&gt; Despite his success in the 1980s, Caine also appeared in some poorly received films such as ''[[Blame It on Rio]]'' (1984), the [[Dick Clement]] and [[Ian La Frenais]] comedy ''[[Water (1985 film)|Water]]'' (1985), the fourth and final film in the [[Jaws (franchise)|''Jaws'' franchise]], ''[[Jaws: The Revenge]]'' (1987), and ''[[Without a Clue]]'' (1988) (portraying [[Sherlock Holmes]]). Caine's commitment to filming ''Jaws: The Revenge'' in the [[Bahamas]] meant that he was unable to receive his Academy Award for ''Hannah and Her Sisters'' in person and Dianne Wiest accepted it on his behalf. Caine said of ''Jaws: The Revenge'', &quot;I have never seen the film, but by all accounts it was terrible. However, I have seen the house that it built, and it is terrific.&quot;&lt;ref&gt;{{cite web|url=http://www.tiscali.co.uk/entertainment/film/biographies/michael_caine_biog/25 |title=Michael Caine Biography |
```

publisher=Tiscali |url-status=dead |archive-url=https://web.archive.org/web/20060910095949/http://www.tiscali.co.uk/entertainment/film/biographies/michael_caine_biog/25 |archive-date=10 September 2006 }}&lt;/ref&gt;&lt;ref&gt;{{cite news|last=Collin|first=Robbie|author-link=Robbie Collin|date=14 March 2013|title=Michael Caine: Extraordinarily good and spectacularly awful|url=https://www.telegraph.co.uk/culture/film/film-news/9926680/Michael-Caine-extraordinarily-good-and-spectacularly-awful.html|archive-url=https://web.archive.org/web/20130314200211/http://www.telegraph.co.uk/culture/film/film-news/9926680/Michael-Caine-Extraordinarily-good-and-spectacularly-awful.html|url-status=dead|archive-date=14 March 2013|newspaper=[[The Daily Telegraph]]|access-date=24 July 2014}}&lt;/ref&gt;

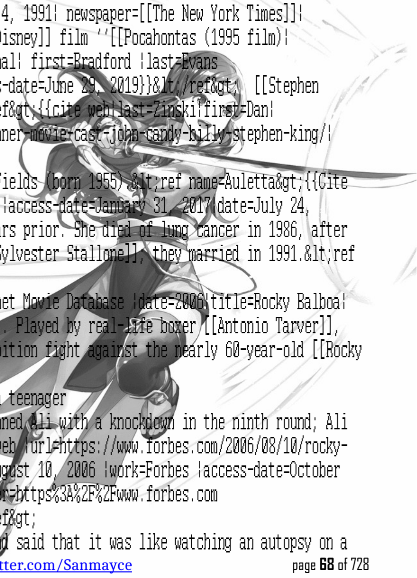
165 Prior to 2000, only seven [[film series]] had grossed over $1&amp;nbsp;billion at the box office: ''[[List of James Bond films|James Bond]]'',&lt;ref&gt;{{cite web |title=Box Office History for James Bond Movies |website=The Numbers |publisher=Nash Information Services |url=https://www.the-numbers.com/movies/series/JamesBond.php |access-date=January 4, 2012}}&lt;/ref&gt; ''[[List of Star Wars films|Star Wars]]'',&lt;ref&gt;{{cite web |title=Box Office History for Star Wars Movies |website=The Numbers |publisher=Nash Information Services |url=https://www.the-numbers.com/movies/series/StarWars.php |access-date=January 4, 2012}}&lt;/ref&gt; ''[[Indiana Jones]]'',&lt;ref&gt;{{cite web |title=Indiana Jones - Worldwide (Unadjusted) |url=https://www.boxofficemojo.com/franchises/chart/?id=indianajones.htm |access-date=January 6, 2012 |website=[[Box Office Mojo]]}}&lt;/ref&gt; ''[[Rocky (film series)|Rocky]]'',&lt;ref&gt;{{Cite news |last=Anderson |first=Dave |date=November 16, 2003 |title=Bayonne Bleeder Throws a Punch at the Italian Stallion |work=[[The New York Times]] |url=https://www.nytimes.com/2003/11/16/sports/othersports/16ANDE.html |access-date=January 4, 2012}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Schneiderman |first=R. M. |date=August 10, 2006 |title=Stallone Settles With The 'Real' Rocky |work=[[Forbes]] |url=https://www.forbes.com/2006/08/10/rocky-stallone-settlement-cx_rs_0810autofacescan04.html |access-date=January 4, 2012}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Poller |first=Kenneth G. |date=November 12, 2003 |title=Charles Wepner v. Sylvester Stallone |publisher=Mango &amp; Iacoviello |url=http://www.mandilaw.com/files/complaint.pdf |access-date=January 4, 2012 |archive-date=October 29, 2013 |archive-url=https://web.archive.org/web/20131029201713/http://www.mandilaw.com/files/complaint.pdf |url-status=dead }}&lt;/ref&gt; ''[[Batman in film|Batman]]'',&lt;ref&gt;{{cite web |title=Batman - Worldwide (Unadjusted) &amp; Batman: Mask of the Phantasm |url=https://www.boxofficemojo.com/franchises/chart/?id=batman.htm |access-date=January 6, 2012 |website=[[Box Office Mojo]]}}&lt;/ref&gt; ''[[Jurassic Park]]'',&lt;ref&gt;{{cite web |title=Jurassic Park - Worldwide (Unadjusted) |url=https://www.boxofficemojo.com/franchises/chart/?id=jurassicpark.htm |access-date=January 6, 2012 |website=[[Box Office Mojo]]}}&lt;/ref&gt; and ''[[List of Star Trek films|Star Trek]]''.&lt;ref&gt;{{cite web |title=Box Office History for Star Trek Movies |website=The Numbers |publisher=Nash Information Services |url=https://www.the-numbers.com/movies/series/StarTrek.php |access-date=January 4, 2012}}&lt;/ref&gt; Since the turn of the century, that number has increased to over a hundred&lt;!-- 101 as of the access date in the reference-102 are listed, but Zootopia is included prematurely (the second film has not yet been released) --&gt;.&lt;ref name=&quot;Numbers franchises&quot;&gt;{{cite web |title=Movie Franchises |url=https://www.the-numbers.com/movies/franchises/sort/World#franchise_overview=l100:od4 |access-date=September 15, 2024 |website=[[The Numbers (website)|The Numbers]] |publisher=Nash Information Services}}&lt;/ref&gt; This is partly due to inflation and market growth, but it is also due to Hollywood's adoption of the [[media franchise|franchise]] model: films that have built-in brand recognition such as being based on a well-known literary source or an established character. The methodology is based on the concept that films associated with things audiences are already familiar with can be more effectively marketed to them, and as such are known as &quot;pre-sold&quot; films within the industry.&lt;ref name=&quot;economist&quot;&gt;{{Cite news |author=The Economist online |date=July 11, 2011 |title=Pottering on, and on |newspaper=[[The Economist]] |url=https://www.economist.com/blogs/dailychart/2011/07/film-franchises |archive-url=https://web.archive.org/web/20110711204558/https://www.economist.com/blogs/dailychart/2011/07/film-franchises |archive-date=July 11, 2011}}&lt;/ref&gt;

166 Longworth notes that the dramatic success of ''Out of Africa'' led to a backlash of critical opinion against Streep in the years that followed, especially as she was now demanding $4 million a picture. Unlike other stars at the time, such as [[Sylvester Stallone]] and [[Tom Cruise]], Streep &quot;never seemed to play herself&quot;, and certain critics felt her technical finesse led people to literally see her acting.{{sfn|Longworth|2013|p=97}} Her next films did not appeal to a wide audience; she co-starred with [[Jack Nicholson]] in the dramas ''[[Heartburn (film)|Heartburn]]'' (1986) and ''[[Ironweed (film)|Ironweed]]'' (1987), in which she sang onscreen for the first time since the &quot;Great Performances&quot; telecast of the Phoenix Theater production of ''Secret Service'' (1977). In ''[[Evil Angels (film)|Evil Angels]]''{{Efn|The film was released outside Australia and New Zealand as ''A Cry in the Dark''.}} (1988), she played [[Lindy Chamberlain]], an Australian woman who had been convicted of the [[Death of Azaria Chamberlain|murder of her infant daughter]] despite claiming that the baby had been taken by a [[dingo]]. Filmed in Australia, Streep won the [[Australian Film Institute Award for Best Actress in a Leading Role]],{{sfn|Waldo|2006|p=209}}{{sfn|Speed|Wilson|1989|p=38|ps=. &quot;Meryl Streep, with black hair and a convincing Aussie accent, is outstanding as Mrs Chamberlain.&quot;}}{{sfn|Eberwein|2010|p=217}} a [[Prix d'interprétation féminine|Best Actress]] at the [[Cannes Film Festival]], and the [[New York Film Critics Circle Award for Best Actress]].{{Sfn|Eberwein|2010|p=221}} Streep has said of developing the Australian accent in the film: &quot;I had to study a little bit for Australian because it's not dissimilar [to&lt;!-- not &quot;from&quot; (UK usage) --&gt; American], so it's like coming from Italian to Spanish. You get a little mixed up.&quot;&lt;ref name=&quot;Brockes06&quot;/&gt; Vincent Canby of ''The New York Times'' referred to her performance as &quot;another stunning performance&quot;, played with &quot;the kind of virtuosity that seems to re-define the possibilities of screen acting&quot;.&lt;ref&gt;{{cite news|author=Canby, Vincent|author-link=Vincent Canby|url=https://www.nytimes.com/movie/review?res=940DE1DF1039F932A25752C1A96E948260|title=A Cry in the Dark|work=The New York Times|date=November 11, 1988|access-date=July 2, 2015|url-status=live|archive-url=https://web.archive.org/web/20150704073433/http://www.nytimes.com/movie/review?res=940DE1DF1039F932A25752C1A96E948260|archive-date=July 4, 2015}}&lt;/ref&gt;

167 A spin-off from increased tourism has been the buying of often enormous tracts of land by foreigners, often as a prestige purchase rather than for agriculture. Buyers have included [[Sylvester Stallone]], [[Ted Turner]], and [[Christopher Lambert]], and most notably [[Luciano Benetton]], Patagonia's largest landowner.&lt;ref name=&quot;time out&quot;/&gt; His &quot;Compañia de Tierras Sud&quot; has brought new techniques to the ailing sheep-rearing industry and sponsored museums and community facilities, but has been controversial particularly for its treatment of local Mapuche communities.&lt;ref&gt;[http://www.mapuche-nation.org/english/main/benetton/main/info.htm 'The Invisible Colors of Benetton'], Mapuche International Link, accessed 2006-08-11&lt;/ref&gt;

168 Candy was in talks to portray Ignatius J. Reilly in a now-shelved film adaptation of [[John Kennedy Toole]]'s [[Pulitzer Prize]]-winning novel ''[[A Confederacy of Dunces]]''.&lt;ref&gt;{{cite magazine| last=Fretts| first=Bruce| title=''A Confederacy of Dunces'' celebrates its 20th anniversary| magazine=[[Entertainment Weekly]]| url=http://www.ew.com/article/2000/05/19/confederacy-dunces-celebrates-its-20th-anniversary| date=May 19, 2000| access-date=July 26, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine| last=Hyman| first=Peter| title=A Conspiracy of Dunces: Will John Kennedy Toole's comic masterpiece ever reach the big screen?| date=December 14, 2006| magazine=[[Slate (magazine)|Slate]]| url=http://www.slate.com/articles/news_and_politics/hollywood/2006/12/a_conspiracy_of_dunces.html| access-date=July 26, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite journal| last=Evans| first=Bradford| title=10 Actors Who Came Close to Starring in 'A Confederacy of Dunces'| journal=[[Splitsider]]| url=http://splitsider.com/2013/03/8-actors-who-came-close-to-starring-in-a-confederacy-of-dunces/| date=March 28, 2013| access-date=July 26, 2015}}&lt;/ref&gt; He had also expressed interest in portraying [[Atuk]] in a film adaptation of [[Mordecai Richler]]'s ''[[The Incomparable Atuk]]'' and [[Roscoe Arbuckle|Roscoe &quot;Fatty&quot; Arbuckle]] in a biopic based on the [[silent film]] comedian's life.&lt;ref&gt;{{cite news| last=Raouf| first=Neda| title=The 'Atuk' Curse| newspaper=[[Los Angeles Times]]| url=https://www.latimes.com/archives/la-xpm-1999-feb-21-tm-10276-story.html| date=February 21, 1999| access-date=August 6, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite news| last=Bovsun| first=Mara| title=Roscoe 'Fatty' Arbuckle, acquitted for murder of Virginia Rappe in 1922, never recovered from all the bad press| newspaper=[[New York Daily News]]| url=http://www.nydailynews.com/news/justice-story/roscoe-fatty-arbuckle-acquitted-murder-virginia-rappe-1922-recovered-bad-press-article-1.1149824| date=September 1, 2012| access-date=August 12, 2015}}&lt;/ref&gt; These three shelved projects have been alleged as cursed because Candy, [[John Belushi]], [[Sam Kinison]], and [[Chris Farley]] were each attached to all three roles; they all died before they could make any of these films.&lt;ref name=split /&gt;&lt;ref&gt;{{cite news| last=Evans| first=Bradford| title=The Lost Roles of John Belushi| journal=Splitsider| url=http://splitsider.com/2011/03/the-lost-roles-of-john-belushi/| date=March 3, 2011| access-date=August 12, 2015| archive-url=https://web.archive.org/web/20180517201328/http://splitsider.com/2011/03/the-lost-roles-of-john-belushi/| archive-date=May 17, 2018| url-status=dead}}&lt;/ref&gt; Candy was originally considered to play [[Alec Guinness]]'s role in the [[Last Holiday (2006 film)|remake]] of the 1950 film ''[[Last Holiday (1950 film)|Last Holiday]]'', with [[Carl Reiner]] directing.&lt;ref&gt;{{cite news| last=Siskel| first=Gene| title=Sour Movies Keep Candy Just Short Of Sweet Success| newspaper=[[Chicago Tribune]]| url=https://www.chicagotribune.com/1986/03/30/sour-movies-keep-candy-just-short-of-sweet-success/| date=March 30, 1986| access-date=July 26, 2015}}&lt;/ref&gt; Eventually the role was played by [[Queen Latifah]] in a [[Last Holiday (2006 film)|loose remake]] released in 2006.&lt;ref name=split&gt;{{cite journal|last=Evans|first=Bradford|title=The Lost Roles of John Candy| journal=Splitsider| url=http://splitsider.com/2011/06/the-lost-roles-of-john-candy/| date=June 2, 2011| access-date=July 26, 2015| archive-url=https://web.archive.org/web/20180520084658/http://splitsider.com/2011/06/the-lost-roles-of-john-candy/| archive-date=May 20, 2018| url-status=dead}}&lt;/ref&gt; Candy was also slated to collaborate with John Hughes again in a comedy opposite [[Sylvester Stallone]], titled ''Bartholomew vs. Neff''. Candy and Stallone were to have portrayed feuding neighbors.&lt;ref&gt;{{cite news| title=SHORT TAKES : Stallone in Line for Comedy Role| date=July 30, 1990| newspaper=Los Angeles Times| url=https://www.latimes.com/archives/la-xpm-1990-07-30-ca-997-story.html| access-date=August 19, 2018}}&lt;/ref&gt;&lt;ref&gt;{{cite news| last=Carter| first=Bill| title=Him Alone| date=August 4, 1991| newspaper=[[The New York Times]]| url=https://www.nytimes.com/1991/08/04/magazine/him-alone.html| access-date=August 19, 2018}}&lt;/ref&gt; In the animated [[The Walt Disney Company|Disney]] film ''[[Pocahontas (1995 film)|Pocahontas]]'', the role of Redfeather the Turkey was written for him, but was subsequently cut from the film after his death.&lt;ref&gt;{{cite journal| first=Bradford |last=Evans |url=http://www.vulture.com/2011/06/the-lost-roles-of-john-candy.html |title=The Lost Roles of John Candy |journal=Vulture |date=June 2, 2011 |access-date=June 29, 2019}}&lt;/ref&gt; [[Stephen King]] reportedly wanted Candy to portray Billy Halleck in the [[Thinner (film)|film adaptation]] of his novel ''[[Thinner (novel)|Thinner]]''.&lt;ref&gt;{{cite web|last=Zinski|first=Dan|title=Stephen King Wanted John Candy To Play The Lead In One Of The Worst Adaptations Of His Work|website=Screen Rant|url=https://screenrant.com/thinner-movie-cast-john-candy-billy-stephen-king/|accessdate=January 22, 2024}}&lt;/ref&gt;

169 He married three times. After graduating from law school, Fields married his college sweetheart, Amy Markson, with whom he had one son, James Elder Fields (born 1955).&lt;ref name=Auletta&gt;{{Cite magazine|first=Ken|last=Auletta|url=https://www.newyorker.com/magazine/2006/07/24/hollywood-ending|title=Hollywood Ending|magazine=[[The New Yorker]]|access-date=January 31, 2017|date=July 24, 2016}}&lt;/ref&gt; In 1960, he married fashion model Lydia Minevitch (daughter of musician [[Borrah Minevitch]]) whose divorce he had handled two years prior. She died of lung cancer in 1986, after 27 years of marriage. He met his third wife, art expert [[Barbara Guggenheim]], after she hired him to defend her when, in 1989, she was sued by [[Sylvester Stallone]]; they married in 1991.&lt;ref name=Auletta /&gt;



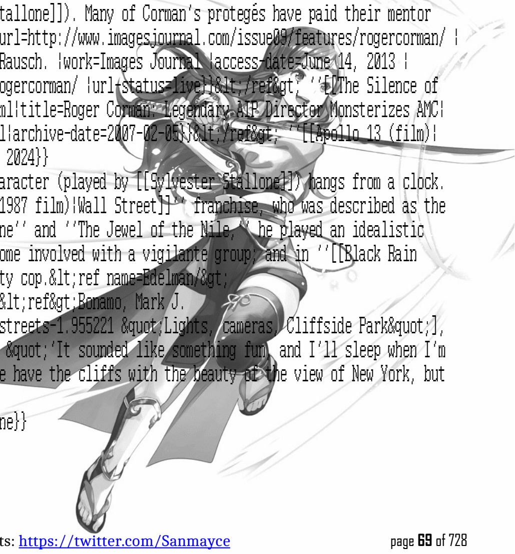
170 * Mason &quot;The Line&quot; Dixon is a leading character in ''[[Rocky Balboa (film)|Rocky Balboa]]'' (2006),&lt;ref&gt;{{cite web|website=The Internet Movie Database |date=2006|title=Rocky Balboa|url=https://www.imdb.com}}&lt;/ref&gt; the sixth film in the [[Rocky (film series)|Rocky franchise]], directed by and starring [[Sylvester Stallone]]. Played by real-life boxer [[Antonio Tarver]], Dixon is the current [[World Heavyweight Boxing Champion]] who is ridiculed for having never fought a real contender, and who thus agrees to an exhibition fight against the nearly 60-year-old [[Rocky Balboa]].&lt;ref&gt;{{cite news|work=Empire Magazine |title=Film Review|date=2007}}&lt;/ref&gt;

171 * [[Sylvester Stallone]]: award-winning actor and director who attended Charlotte Hall Military Academy and boarding school in St. Mary's County as a teenager

172 Ali's next opponents included [[Chuck Wepner]], [[Ron Lyle]], and [[Joe Bugner]]. Wepner, a journeyman known as &quot;The Bayonne Bleeder&quot;, stunned Ali with a knockdown in the ninth round; Ali would later say he tripped on Wepner's foot. The fight inspired [[Sylvester Stallone]] to create the acclaimed film ''[[Rocky]]''.&lt;ref&gt;{{Cite web |url=https://www.forbes.com/2006/08/10/rocky-stallone-settlement-cx_rs_0810autofacescan04.html#34f600e166bd |title=Stallone Settles With The 'Real' Rocky |first=R. M. |last=Schneiderman |date=August 10, 2006 |work=Forbes |access-date=October 16, 2019 |archive-date=October 31, 2023 |archive-url=https://web.archive.org/web/20231031224107/https://e.infogram.com/_/PGbi4jGUl7iSfajoEZwo?referrer=https%3A%2F%2Fwww.forbes.com%2F2006%2F08%2F10%2Frocky-stallone-settlement-cx_rs_0810autofacescan04.html%3Fsh%3D29658a0a66bd&amp;src=embed#34f600e166bd |url-status=live }}&lt;/ref&gt;

173 Giachetti called the fight &quot;awful...the worst sports event I ever had to cover&quot;. Actor [[Sylvester Stallone]] was ringside for the fight and said that it was like watching an autopsy on a

```
man who is still alive.&lt;ref name=&quot;Hauser 2004&quot; /&gt; The Holmes fight is said to have contributed to Ali's [[Parkinson's syndrome]].&lt;ref&gt;{{cite news |url=https://www.nytimes.com/2009/10/27/arts/television/27muhammad.html |work=The New York Times |title=Boxing King Casts His Shadow, Even at Time of Defeat |date=October 26, 2009 |access-date=March 5, 2012 |first1=Mike |last1=Hale |archive-date=March 10, 2012 |archive-url=https://web.archive.org/web/20120310053609/http://www.nytimes.com/2009/10/27/arts/television/27muhammad.html? |url-status=live }}&lt;/ref&gt;&lt;ref&gt;[https://www.cbc.ca/sports/muhammad-ali-doctor-doubts-parkinsons-1.3619442 ''AP''] {{Webarchive|url=https://web.archive.org/web/20201112024309/https://www.cbc.ca/sports/muhammad-ali-doctor-doubts-parkinsons-1.3619442 |date=November 12, 2020 }} &quot;Muhammad Ali's doctor doubts boxing led to Parkinson's&quot;, ''Associated Press'' via ''CBC'', June 6, 2016. Retrieved September 1, 2018.&lt;/ref&gt;&lt;ref name=&quot;nytimes&quot; /&gt; Despite pleas to definitively retire, Ali fought one last time on [[Muhammad Ali vs. Trevor Berbick|December 11, 1981]], in Nassau, [[Bahamas]], against [[Trevor Berbick]], losing a ten-round decision.&lt;ref&gt;{{cite news |title=Ali to try again? |url=https://news.google.com/newspapers?nid=1870&amp;dat=19810816&amp;id=9_EwAAAAIBAJ&amp;pg=2603,113296&amp;hl=en |newspaper=The Daytona Beach Sunday News-Journal |agency=Associated Press |date=August 16, 1981 |access-date=June 4, 2016 |archive-date=August 28, 2021 |archive-url=https://web.archive.org/web/20210828083317/https://news.google.com/newspapers?nid=1870&amp;dat=19810816&amp;id=9_EwAAAAIBAJ&amp;pg=2603%2C113296&amp;hl=en |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite news |title=It's all over for Ali after loss |url=https://news.google.com/newspapers?nid=2199&amp;dat=19811212&amp;id=XKUyAAAAIBAJ&amp;pg=3700,2405394&amp;hl=en |newspaper=Lawrence Journal-World |agency=Associated Press |date=December 12, 1981 |access-date=June 4, 2016 |archive-date=September 29, 2021 |archive-url=https://web.archive.org/web/20210929102632/https://news.google.com/newspapers?nid=2199&amp;dat=19811212&amp;id=XKUyAAAAIBAJ&amp;pg=3700,2405394&amp;hl=en |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite magazine |first=William |last=Nack |author-link=William Nack |title=Not with a bang but a whisper |url=https://www.si.com/vault/1981/12/21/826243/not-with-a-bang-but-a-whisper-after-losing-to-trevor-berbick-a-subdued-muhammad-ali-softly-admitted-that-his-illustrious-career-had-come-to-an-end |magazine=Sports Illustrated |date=December 21, 1981 |access-date=June 4, 2016 |archive-date=June 9, 2016 |archive-url=https://web.archive.org/web/20160609212952/http://www.si.com/vault/1981/12/21/826243/not-with-a-bang-but-a-whisper-after-losing-to-trevor-berbick-a-subdued-muhammad-ali-softly-admitted-that-his-illustrious-career-had-come-to-an-end |url-status=live }}&lt;/ref&gt;
174 * [[Brigitte Nielsen]] (1963-), actress, former wife of [[Sylvester Stallone]]
175 *''Blood Oath'' by [[Christopher Farnsworth]] (2010).&lt;ref&gt;{{cite web|url=http://www.dreadcentral.com/news/37567/blood-oath-a-political-thriller-novel-with-vampires|title=Sylvester Stallone Tracking Creatures in the Arctic Circle in Hunter - Dread Central|website=www.dreadcentral.com|date=18 September 2012 |access-date=4 May 2018|url-status=live|archive-url=https://web.archive.org/web/20121014005550/http://www.dreadcentral.com/news/37567/blood-oath-a-political-thriller-novel-with-vampires|archive-date=14 October 2012}}&lt;/ref&gt;
176 Johnson continued taking paying fights for many years, and operated several other businesses, including lucrative endorsement deals. He died in a car crash in 1946 at the age of 68.&lt;ref name=&quot;Internet Archive&quot;&gt;{{cite web|url=https://www.usnews.com/usnews/news/articles/050117/17burns.peo.htm |title=U.S. News &amp; World Report 'Two champs meet' |publisher=[[Internet Archive]] |access-date=February 14, 2015 |url-status=dead |archive-url=https://web.archive.org/web/20131111070334/http://www.usnews.com/usnews/news/articles/050117/17burns.peo.htm |archive-date=November 11, 2013 }}&lt;/ref&gt; He is buried at [[Graceland Cemetery]] in Chicago. In 2018, Johnson, with some instigation from actor [[Sylvester Stallone]], was formally pardoned by U.S. president [[Donald Trump]].&lt;ref&gt;{{Cite news|last1=Eligon|first1=John|last2=Shear|first2=Michael D.|date=2018-05-24|title=Trump Pardons Jack Johnson, Heavyweight Boxing Champion (Published 2018)|language=en-US|work=The New York Times|url=https://www.nytimes.com/2018/05/24/sports/jack-johnson-pardon-trump.html|access-date=2021-02-25|issn=0362-4331}}&lt;/ref&gt;
177 President [[Donald Trump]] granted Johnson a posthumous [[presidential pardon]] after recurring proposals to grant one had not been acted on by previous administrations. In April 2018, Trump announced that he was considering granting a full pardon to Johnson after speaking with a World Boxing Council committee,{{Failed verification|date=February 2023}} on the advice of actor [[Sylvester Stallone]].&lt;ref name=Politico20180421&gt;{{cite web |last1=Griffiths |first1=Brent D. |title=Trump says he's 'considering' a pardon for boxer Jack Johnson |url=https://www.politico.com/story/2018/04/21/trump-pardon-jack-johnson-boxer-544930 |publisher=Politico |archive-url=https://web.archive.org/web/20180422031538/https://www.politico.com/story/2018/04/21/trump-pardon-jack-johnson-boxer-544930 |archive-date=April 22, 2018 |date=April 21, 2018 | url-status=live }}&lt;/ref&gt; Trump pardoned Johnson on May 24, 2018, 105 years after his conviction, during a ceremony which included special guests Sylvester Stallone (actor), [[Deontay Wilder]] (then current WBC Champion), [[Lennox Lewis]] (WBC Former Champion), [[Mauricio Sulaiman]] (WBC President), Linda Bell Haywood (Johnson's great-great niece), and Hector Sulaiman (President of the Board of Advisors of Scholas Occurrentes).&lt;ref name=&quot;:0&quot;&gt;{{cite web | url=https://www.nytimes.com/2018/05/24/sports/jack-johnson-pardon-trump.html |title=Trump Pardons Jack Johnson, Heavyweight Boxing Champion |last1=Eligon|first1=John|last2=Shear|first2=Michael D.| date=May 24, 2018 |website=The New York Times|access-date=May 26, 2018}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.nytimes.com/2018/05/24/sports/jack-johnson-racism.html|title=Missed in Coverage of Jack Johnson, the Racism Around Him |last1=Eligon |first1=John|last2=Thorpe|first2=Brandon K.|date=May 24, 2018|website=The New York Times|access-date=May 26, 2018}}&lt;/ref&gt;
178 [[Harrison Ford]] was Spielberg's original choice to play Eddie Valiant, but his price was too high.&lt;ref&gt;{{cite web|url=https://uproxx.com/movies/harrison-ford-lost-roles/|title='Who Framed Roger Rabbit?' And Six Other Big Roles Harrison Ford Missed Out On|date=July 13, 2015|access-date=May 26, 2017|archive-date=September 13, 2017|archive-url=https://web.archive.org/web/20170913231452/http://uproxx.com/movies/harrison-ford-lost-roles/|url-status=live}}&lt;/ref&gt; [[Chevy Chase]] was the second choice, but he was not interested.&lt;ref name=&quot;mentalfloss.com&quot; /&gt; [[Bill Murray]] was also considered for the role, but due to his idiosyncratic method of receiving offers for roles, Murray missed out on it.&lt;ref&gt;{{cite web|url=https://www.huffingtonpost.com/john-farr/bill-murray-and-the-roles_b_5850434.html|title=Bill Murray and the Roles That Got Away|first=John|last=Farr|date=September 19, 2014|access-date=May 26, 2017|work=[[HuffPost]]|archive-date=June 19, 2015|archive-url=https://web.archive.org/web/20150619004834/http://www.huffingtonpost.com/john-farr/bill-murray-and-the-roles_b_5850434.html|url-status=live}}&lt;/ref&gt; [[Eddie Murphy]] reportedly turned down the role as he misunderstood the concept of toons and humans co-existing; he later regretted this decision.&lt;ref name=&quot;Evans&quot;&gt;{{cite web|last=Evans|first=Bradford|title=The Lost Roles of Eddie Murphy|date=April 7, 2011|publisher=[[Splitsider]]|url=https://www.vulture.com/2011/04/the-lost-roles-of-eddie-murphy/|access-date=July 18, 2015|archive-url=https://web.archive.org/web/20150723072044/http://splitsider.com/2011/04/the-lost-roles-of-eddie-murphy/|archive-date=July 23, 2015|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last1=Fallon |first1=Jimmy |title=Eddie Murphy Confirms Rumors and Stories About Prince, Ghostbusters and More |url=https://www.youtube.com/watch?v=CABVnXoe9gw |website=YouTube |date=December 20, 2019 |access-date=December 23, 2019 |archive-date=December 23, 2019 |archive-url=https://web.archive.org/web/20191223032637/https://www.youtube.com/watch?v=CABVnXoe9gw |url-status=live }}&lt;/ref&gt; [[Robin Williams]], [[Robert Redford]], [[Jack Nicholson]], [[Sylvester Stallone]], [[Edward James Olmos]], [[Wallace Shawn]], [[Ed Harris]], [[Charles Grodin]] and [[Don Lane]] were also considered for the role.&lt;ref name=&quot;mentalfloss.com&quot;&gt;{{cite web|url=https://www.mentalfloss.com/article/62910/15-things-you-might-not-know-about-who-framed-roger-rabbit|title=15 Things You Might Not Know About ''Who Framed Roger Rabbit'' |work=[[Mental Floss]] |date=April 10, 2015|access-date=May 26, 2017|archive-date=June 4, 2017|archive-url=https://web.archive.org/web/20170604194418/http://mentalfloss.com/article/62910/15-things-you-might-not-know-about-who-framed-roger-rabbit|url-status=live}}&lt;/ref&gt; Ultimately, [[Bob Hoskins]] was chosen by Spielberg because of his acting skill and because Spielberg believed he had a hopeful demeanor and he looked like he belonged in that era.&lt;ref name=&quot;:1&quot;&gt;{{Cite web|date=2018-06-13|title=Who Framed Roger Rabbit: A Look at How the Cartoons Came to Life|url=https://www.closerweekly.com/posts/who-framed-roger-rabbit-161764/|access-date=2022-02-15|website=Closer Weekly|language=en-US}}&lt;/ref&gt;
179 As a Hollywood screenwriter, Kamen was mentored by [[Frank Price]] who told him that producer [[Jerry Weintraub]] had optioned a news article about the young child of a single mother who had earned a black belt to defend himself against the neighborhood bullies. Kamen then combined his own life story with the news article and used both to create the screenplay for ''The Karate Kid''.&lt;ref name=oralhistory/&gt; Additionally, given [[John G. Avildsen]]'s involvement with both films, [[Sylvester Stallone]] often joked with Kamen that the writer had &quot;ripped off&quot; the ''[[Rocky (film series)|Rocky]]'' films with ''The Karate Kid''.&lt;ref name=oralhistory/&gt;
180 A soundtrack album was released in 1984 by [[Casablanca Records]] containing many of the contemporary songs featured in the film. Of particular note is [[Joe Esposito (singer)|Joe Esposito]]'s &quot;[[You're the Best]]&quot;, featured during the tournament [[Montage sequence|montage]] near the end of the first film. Originally written for ''[[Rocky III]]'' (1982), &quot;You're the Best&quot; was rejected by [[Sylvester Stallone]] in favor of Survivor's hit song &quot;[[Eye of the Tiger]]&quot;.&lt;ref name=making/&gt; Coincidentally, Survivor also performed the main theme (&quot;The Moment of Truth&quot; Music &amp; Lyrics: Bill Conti, Dennis Lambert, Peter Beckett) for the movie.
181 Actors who obtained their career breaks working for Corman include [[Jack Nicholson]], [[Peter Fonda]], [[Bruce Dern]], [[Charles Bronson]], [[Todd Field]]&lt;ref&gt;{{Cite web |url=http://www.tcm.com/tcmdb/person/61501%7C175120/Todd-Field/biography.html |publisher=Turner Classic Movies |date=November 29, 2015 |title=Todd Field Biography |access-date=November 29, 2015 | archive-date=December 8, 2015 |archive-url=https://web.archive.org/web/20151208051221/http://www.tcm.com/tcmdb/person/61501%7C175120/Todd-Field/biography.html |url-status=live}}&lt;/ref&gt; [[Michael McDonald (actor)|Michael McDonald]], [[Dennis Hopper]], [[Tommy Lee Jones]], [[Talia Shire]], [[Sandra Bullock]], [[Robert De Niro]], and [[David Carradine]], who received one of his first starring film roles in the Corman-produced ''[[Boxcar Bertha]]'' (1972) and went on to star in ''[[Death Race 2000]]'' (along with [[Sylvester Stallone]]). Many of Corman's protegés have paid their mentor homage by awarding him cameos in films, such as in ''[[The Godfather Part II]]'',&lt;ref name=&quot;imagesjournal&quot;&gt;{{cite web |url=http://www.imagesjournal.com/issue09/features/rogercorman/ | title=Roger Corman on The Blair Witch Project and why Mean Streets would have made a great blaxploitation film. Interview by Andrew J. Rausch. |work=Images Journal |access-date=June 14, 2013 | archive-date=September 1, 2017 |archive-url=https://web.archive.org/web/20170901055248/https://www.imagesjournal.com/issue09/features/rogercorman/ |url-status=live}}&lt;/ref&gt; ''[[The Silence of the Lambs (film)|The Silence of the Lambs]]'',&lt;ref&gt;{{cite web|url=http://www.filmfax.com/archives/amc_monsterized/roger_corman.html|title=Roger Corman: Legendary AIP Director Monsterizes AMC|url-status=dead|archive-url=https://web.archive.org/web/20070205231122/http://www.filmfax.com/archives/amc_monsterized/roger_corman.html|archive-date=2007-02-05}}&lt;/ref&gt; ''[[Apollo 13 (film)|Apollo 13]]'',&lt;ref name=&quot;EW&quot;/&gt; and as recently as the Demme film ''[[Rachel Getting Married]]'' (2008).{{fact|date=June 2024}}
182 *The 1991 comedy film ''[[Oscar (1991 film)|Oscar]]'' paid a direct homage to the scene, recreating it on its poster, where the main character (played by [[Sylvester Stallone]]) hangs from a clock.
183 Conversely, Douglas also played powerful characters with dominating personalities, including [[Gordon Gekko]], in the ''[[Wall Street (1987 film)|Wall Street]]'' franchise, who was described as the &quot;greedy yuppie personification of the Me generation,&quot; and says &quot;greed is good&quot; in the movie; in ''Romancing the Stone'' and ''The Jewel of the Nile,'' he played an idealistic soldier of fortune; in ''[[The Star Chamber]]'' (1983), he was a court judge fed up with an inadequate legal system, leading him to become involved with a vigilante group; and in ''[[Black Rain (American film)|Black Rain]]'' (1989), he proved he could also play a [[Sylvester Stallone|Stallone]]-like action hero as a New York City cop.&lt;ref name=Edelman/&gt;
184 Scenes from ''[[Cop Land]]'', directed by James Mangold and starring Sylvester Stallone, Harvey Keitel and Ray Liotta were filmed here.&lt;ref&gt;Bonamo, Mark J. [https://web.archive.org/web/20160919010908/http://www.northjersey.com/arts-and-entertainment/indie-filmmakers-pursue-dreams-on-bergen-streets-1.955221 &quot;Lights, cameras, Cliffside Park&quot;], ''Cliffside Park Citizen'', June 11, 2010, backed up by the [[Internet Archive]] as of September 19, 2016. Accessed September 10, 2017. &quot;'It sounded like something fun, and I'll sleep when I'm dead,' said Snyder. 'They filmed ''Big'' and ''Cop Land'' in Cliffside Park, and I think Bergen County is an excellent place to film. We have the cliffs with the beauty of the view of New York, but it's still a very suburban place. This is perfect Americana.'&quot;&lt;/ref&gt;
185       <text bytes="45441" sha1="tuqnm5w1k4014svy22sb6nh96xs1rb6" xml:space="preserve">{{Short description|1985 film by Sylvester Stallone}}
186 | director       = [[Sylvester Stallone]]
187 | writer         = Sylvester Stallone
188 | based_on       = {{Based on|[[List of Rocky characters|Characters]]|Sylvester Stallone}}
189 * Sylvester Stallone
```



190 '''''Rocky IV''''' is a 1985 American [[Sports film|sports]] [[Drama (film and television)|drama film]] starring, written and directed by [[Sylvester Stallone]].&lt;ref name=&quot;Rocky IV&quot;&gt;{{cite web |url=https://www.tcm.com/tcmdb/title/20886/Rocky-IV/ |title=Rocky IV |work=TCM database |publisher=[[Turner Classic Movies]]|access-date=February 28, 2016}}&lt;/ref&gt; The film is the sequel to ''[[Rocky III]]'' (1982) and the fourth installment in the [[Rocky (franchise)|''Rocky'' film series]]. It also stars [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], [[Brigitte Nielsen]], and [[Dolph Lundgren]]. In the film, [[Rocky Balboa]] (Stallone) confronts [[Ivan Drago]] (Lundgren), a Soviet boxer responsible for another personal tragedy in Balboa's life.

191 * [[Sylvester Stallone]] as [[Rocky Balboa|Robert &quot;Rocky&quot; Balboa]], &quot;The Italian Stallion&quot;: The reigning two-time Heavyweight Champion of the World.

192 Sylvester Stallone has stated that the original punching scenes filmed between him and Dolph Lundgren in the first portion of the fight are completely authentic. Stallone wanted to capture a realistic scene and Lundgren agreed that they would engage in legitimate sparring. One particularly forceful punch to Stallone's chest slammed his heart against his breastbone, causing the heart to swell. Stallone, suffering from labored breathing and a blood pressure over 200, was flown from the set in [[Vancouver, British Columbia]] to [[Saint John's Health Center|Saint John's Regional Medical Center]] in [[Santa Monica]], [[California]] and was forced into intensive care for eight days. Stallone later commented that he believed Lundgren had the athletic ability and talent to fight in the professional heavyweight division of boxing.&lt;ref name=&quot;AICN&quot;&gt;{{cite web |url=https://legacy.aintitcool.com/node/30932 |title=Stallone Interview With Ain't It Cool News |publisher=AICN |access-date=January 25, 2007}}&lt;/ref&gt; Producer Winkler describes the exact same event in his autobiography, observing not Lundgren, rather, &quot;Sly took a punch from a [[stand-in]] fighter and ended up in the emergency room with his blood pressure dangerously high.&quot;&lt;ref&gt;{{Cite book |last=Winkler |first=Irwin |title=A Life in Movies |publisher=Abrams Press |year=2019 |isbn=9781683355281 |location=New York|pages=111}}&lt;/ref&gt;

193 Songs from the movie include &quot;[[Living in America (James Brown song)|Living in America]]&quot; by [[James Brown]], as well as music by [[John Cafferty]] (&quot;[[Heart's on Fire (John Cafferty song)|Heart's on Fire]]&quot;, featuring Vince DiCola), [[Survivor (band)|Survivor]], [[Kenny Loggins]], and [[Robert Tepper]]. Four of these songs became U.S. chart hits, two of which reached the Top Five. [[Go West (band)|Go West]] wrote &quot;One Way Street&quot; for the movie by request of Sylvester Stallone. [[Europe (band)|Europe]]'s hit &quot;[[The Final Countdown (song)|The Final Countdown]]&quot;, written earlier in the decade by lead singer [[Joey Tempest]], is often incorrectly stated as being featured in the film due to its similarity to DiCola's &quot;Training Montage.&quot;{{citation needed|date=December 2014}} However, Europe's track was not released as a single until late 1986, after ''Rocky IV''&lt;nowiki&gt;'&lt;/nowiki&gt;s release.

194 In August 2020, Stallone announced that a [[director's cut]] edition of the film would be released to commemorate the film's 35th anniversary.&lt;ref&gt;{{cite web |url=https://screenrant.com/rocky-4-directors-cut-footage-drago-creed-sylvester-stallone/ |title=Sylvester Stallone Releases New Footage of Rocky 4 Director's Cut |first=Mike |last=Jones |website=[[Screen Rant]] |date=September 10, 2020 |access-date=November 2, 2020}}&lt;/ref&gt; Approximately 38 minutes of previously unreleased footage consisting of both new scenes and alternate takes was added to the film, including significant extensions of both fight scenes and the [[Apollo Creed]] funeral scene, in addition to a lengthier recap of ''Rocky III'' at the beginning of the film. A significant amount of original footage was removed or replaced to make way for these additions, as the director's cut runs 93 minutes compared to the original cut's 91 minutes.&lt;ref name=movie-cen/&gt;&lt;ref&gt;{{cite web |url=https://www.empireonline.com/movies/reviews/rocky-iv-rocky-vs-drago-the-ultimate-director-cut |title=Sylvester Stallone Says Director's Cut of 'Rocky IV' Coming for Film's Anniversary |first=Terri |last=White |website=Empireonline.com |date=October 18, 2021 |access-date=October 18, 2021}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |date=October 18, 2021 |title=Rocky IV: Rocky vs. Drago - The Ultimate Director's Cut Review |url=https://www.empireonline.com/movies/reviews/rocky-iv-rocky-vs-drago-the-ultimate-director-cut/|access-date=October 18, 2021 |website=Empireonline.com |language=en-US}}&lt;/ref&gt;

195 One major cut was that of the scenes featuring Paulie's robot. &quot;The robot is going to the junkyard forever, no more robot,&quot; Stallone commented.&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/heat-vision/sylvester-stallone-says-directors-cut-of-rocky-iv-coming-for-films-anniversary |title=Sylvester Stallone Says Director's Cut of 'Rocky IV' Coming for Film's Anniversary |first=Ryan |last=Parker |website=[[The Hollywood Reporter]] |date=August 30, 2020 |access-date=November 2, 2020}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://screenrant.com/rocky-4-directors-cut-paulie-robot-sylvester-stallone/ |title=Rocky IV Director's Cut Won't Include That Ridiculous Robot |first=Humza |last=Hussain |website=Screen Rant |date=August 31, 2020 |access-date=November 2, 2020}}&lt;/ref&gt; Robert Doornick, founder of International Robotics and the voice of the robot, commented that Stallone cut all of the robot scenes in the director's cut to save money on royalty fees that were given to Doornick in the original cut.&lt;ref&gt;{{cite magazine |url=https://www.empireonline.com/movies/news/rocky-iv-robot-creator-speaks-out-directors-cut-exclusive/ |title=Rocky IV: Paulie's Robot Creator Speaks Out About Director's Cut Controversy - Exclusive |first=Ben |last=Travis |magazine=[[Empire (magazine)|Empire]] |date=September 29, 2020 |access-date=November 2, 2020}}&lt;/ref&gt; Scenes with [[Brigitte Nielsen]] playing Ludmilla, Ivan Drago's wife, were also substantially diminished, particularly the scenes where she spoke for her husband at press conferences and her encounter with Apollo Creed's wife in Las Vegas just before the tragic fight between Creed and Drago. Nielsen and Stallone were married in 1985, shortly after the film's premiere, and divorced in 1987, which perhaps explains Stallone's decision to now reduce the role of his ex-wife in the film.&lt;ref&gt;{{Cite web |date=November 18, 2021 |title=Rocky IV |first=Felipe |last=Guerra |url=https://medium.com/fan-fare/every-change-in-the-new-stallones-cut-of-rocky-iv-39232d1b5f2|access-date=November 18, 2021 |website=FanFare}}&lt;/ref&gt; In contrast, additional minor scenes featuring Adrian and Drago's manager Nicolai were added to the movie.&lt;ref name=movie-cen/&gt;

196 The director's cut, titled ''Rocky IV: Rocky vs. Drago - The Ultimate Director's Cut'', had a one-night theatrical release on November 11, 2021, and was released in digital formats the following day.&lt;ref&gt;{{Cite web |date=June 8, 2021 |title=Rocky 4 Director's Cut Gets November 2021 Theatrical Release Date |url=https://screenrant.com/rocky-4-directors-cut-release-date-sylvester-stallone/|access-date=June 8, 2021 |website=ScreenRant |language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=D'Alessandro |first=Anthony |date=September 30, 2021 |title=Sylvester Stallone's Rocky IV: Rocky vs. Drago - The Ultimate Director's Cut Hitting Theaters &amp; Digital |url=https://deadline.com/2021/09/sylvester-stallones-rocky-iv-rocky-vs-drago-ultimate-directors-cut-hitting-theaters-digital-1234847353/|access-date=September 30, 2021 |website=Deadline |language=en-US}}&lt;/ref&gt; Unlike the original cut, the director's cut was distributed through [[United Artists Releasing]]. A &quot;making of&quot; video was released on [[YouTube]] a week prior to the director's cut release.&lt;ref&gt;{{Cite web |last=Stallone |first=Sylvester |date=October 30, 2021 |title=The Making of ROCKY VS. DRAGO by Sylvester Stallone |url=https://www.youtube.com/watch?v=suhqtn_brBs |access-date=May 10, 2023 |website=YouTube |publisher=Sly Stallone Shop}}&lt;/ref&gt;

197 The film won five [[Golden Raspberry Awards]], including [[Golden Raspberry Award for Worst Actor|Worst Actor]] (Sylvester Stallone, along with ''[[Rambo: First Blood Part II]]''), [[Golden Raspberry Award for Worst Director|Worst Director]] (Stallone), [[Golden Raspberry Award for Worst Supporting Actress|Worst Supporting Actress]] (Brigitte Nielsen), [[Golden Raspberry Award for Worst New Star|Worst New Star]] (Nielsen, and also for ''[[Red Sonja (1985 film)|Red Sonja]]'') and [[Golden Raspberry Award for Worst Musical Score|Worst Musical Score]]. It also received nominations for [[Golden Raspberry Award for Worst Picture|Worst Picture]], [[Golden Raspberry Award for Worst Supporting Actress|Worst Supporting Actress]] (Talia Shire), [[Golden Raspberry Award for Worst Supporting Actor|Worst Supporting Actor]] (Burt Young) and [[Golden Raspberry Award for Worst Screenplay|Worst Screenplay]].&lt;ref&gt;{{cite web |url=https://razzies.com/asp/content/XcNewsPlus.asp?cmd=view&amp;articleid=26 |title=1985 Archive |author=Wilson, John |publisher=[[Golden Raspberry Award|Razzies.com]] |date=January 2, 2002 |access-date=October 22, 2012 |url-status=dead |archive-url=https://web.archive.org/web/20141013200532/http://razzies.com/asp/content/XcNewsPlus.asp?cmd=view&amp;articleid=26 |archive-date=October 13, 2014 }}&lt;/ref&gt;

198 A [[novelization]] was published by [[Ballantine Books]] in 1985. Sylvester Stallone was credited as the author.&lt;ref&gt;{{cite book |title=Rocky IV |first=Sylvester |last=Stallone |date=January 1, 1985 |publisher=Ballantine Bks. |oclc=12923839}}&lt;/ref&gt; The novel included some backstory for Drago and his wife. Drago was a former coal miner who had come to the government's attention after being featured in a Party promotional film about mining. Ludmilla, born in [[Kiev]] to a Party official, had begun training to become a swimming Olympic champion when she was nine.

199 [[Sylvester Stallone]], [[Dolph Lundgren]], and [[Brigitte Nielsen]] reprise their roles from ''Rocky IV'' in ''[[Creed II]]'', the sequel to the 2015 film ''[[Creed (film)|Creed]]''. The plot involves Drago's son Viktor fighting [[Adonis Creed]] ([[Michael B. Jordan]]) for his title as a way of his father Ivan regaining some of the prestige he lost after his defeat in ''Rocky IV.''&lt;ref&gt;{{cite web |last1=Grech |first1=Kevin |title=Sylvester Stallone - The making of Rocky IV: Rocky vs. Drago |url=https://www.evolutionofbodybuilding.net/sylvester-stallone-making-of-rocky-iv-rocky-vs-drago/ |website=Evolutionofbodybuilding.net |date=November 13, 2021 |publisher=Evolution of Bodybuilding |access-date=November 13, 2021}}&lt;/ref&gt;

200 {{Sylvester Stallone}}

201 [[Category:Films directed by Sylvester Stallone]]

202 [[Category:Films with screenplays by Sylvester Stallone]]

203 In ''[[Twilight Zone: The Movie]]'' (1983), the scenes of the rest home in the &quot;Kick the Can&quot; segment were filmed at the historic building that is now the Heritage Valley Inn. [[Dolly Parton]] and [[Sylvester Stallone]] filmed scenes for ''[[Rhinestone (film)|Rhinestone]]'' (1984), in Piru. Also ''Silent Night Deadly Night 3'' was filmed here, in 1984.

204 The town served as the fictional San Remos countryside in the movie ''[[Cobra (1986 film)|Cobra]]'', starring [[Sylvester Stallone]], [[Reni Santoni]] and [[Brigitte Nielsen]].

205 * [[Sylvester Stallone]], actor&lt;ref name=&quot;latimes10&quot;&gt;{{cite news|url=https://www.latimes.com/business/realestate/la-fi-hotprop-sylvester-stallone-20110915-story.html|author=Lauren Beale, Los Angeles Times|title=Sylvester Stallone sells waterfront retreat in Thousand Oaks|newspaper=[[Los Angeles Times]]|access-date=November 28, 2017|url-status=live|archive-url=https://web.archive.org/web/20171008031043/http://www.latimes.com/business/realestate/la-fi-hotprop-sylvester-stallone-20110915-story.html|archive-date=October 8, 2017}}&lt;/ref&gt;

206 * [[Don Kernodle]], born in Burlington, five-time NWA champion and tag team partner of Sgt Slaughter; appeared in ''[[Paradise Alley]]'' with [[Sylvester Stallone]]

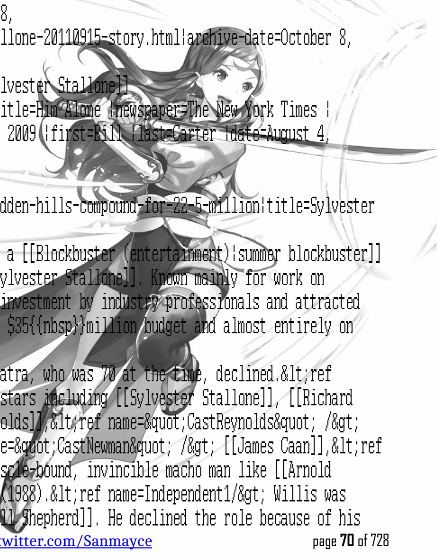
207 * ''Bartholomew Vs. Neff'' - a vehicle that would have starred [[Sylvester Stallone]] and John Candy as feuding neighbors&lt;ref&gt;{{cite news |title=Him Alone |newspaper=The New York Times |url=https://query.nytimes.com/gst/fullpage.html?res=9D0CE6DC143AF937A3575BC0A967958260&amp;sec=&amp;spon=&amp;pagewanted=all |access-date=March 3, 2009 |first=Bill |last=Carter |date=August 4, 1991}}&lt;/ref&gt; (1991)

208 *[[Sylvester Stallone]] (2010)

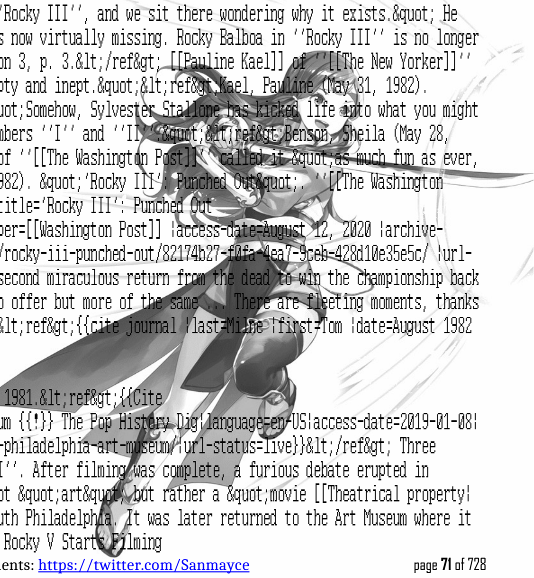
209 *[[Sylvester Stallone]], actor&lt;ref&gt;{{Cite news|url=https://www.latimes.com/business/real-estate/story/2022-12-19/sylvester-stallone-lists-hidden-hills-compound-for-22-5-million|title=Sylvester Stallone lists Hidden Hills compound for $22.5 million|newspaper=Los Angeles Times|date=December 19, 2022|access-date=July 27, 2024}}&lt;/ref&gt;

210 Stuart was hired by [[20th Century Fox]] to adapt Thorp's novel in 1987. His first draft was [[greenlit]] immediately, as the studio was eager for a [[Blockbuster (entertainment)|summer blockbuster]] the following year. The role of McClane was turned down by a host of the decade's most popular actors, including [[Arnold Schwarzenegger]] and [[Sylvester Stallone]]. Known mainly for work on television, Willis was paid $5{{nbsp}}million for his involvement, placing him among Hollywood's highest-paid actors. The deal was seen as a poor investment by industry professionals and attracted significant controversy towards the film prior to its release. Filming took place between November 1987 and March 1988, on a $25{{nbsp}}million to $35{{nbsp}}million budget and almost entirely on location in and around [[Fox Plaza (Los Angeles)|Fox Plaza]] in Los Angeles.

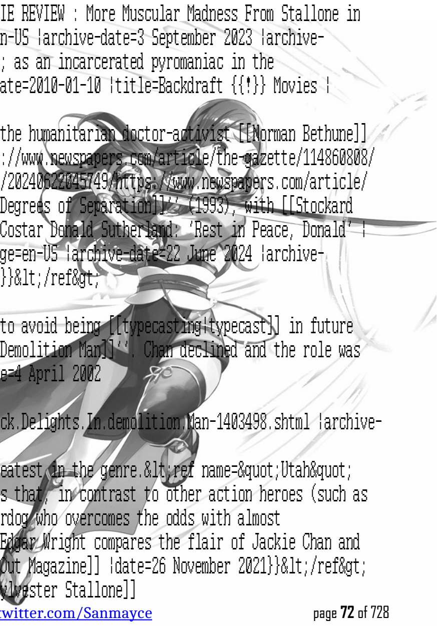
211 As ''Die Hard'' was based on the novel sequel to ''The Detective'' film, the studio was contractually obliged to offer Frank Sinatra the role. Sinatra, who was 70 at the time, declined.&lt;ref name=Independent1/&gt;&lt;ref name=&quot;VanityFairat30&quot;/&gt;&lt;ref name=&quot;EmpireRetro&quot;/&gt; The role was offered to various major stars including [[Sylvester Stallone]], [[Richard Gere]], [[Clint Eastwood]], [[Harrison Ford]],&lt;ref name=&quot;VultureRetro&quot; /&gt;&lt;ref name=&quot;DevOtherActors&quot; /&gt; [[Burt Reynolds]],&lt;ref name=&quot;CastReynolds&quot; /&gt; [[Nick Nolte]], [[Mel Gibson]], [[Don Johnson]], [[Richard Dean Anderson]],&lt;ref name=&quot;Independent1&quot; /&gt; [[Paul Newman]],&lt;ref name=&quot;CastNewman&quot; /&gt; [[James Caan]],&lt;ref name=&quot;DailyBeastCaan&quot; /&gt; and [[Al Pacino]].&lt;ref name=&quot;AlPacino&quot;/&gt; The prevailing action archetype of the era was a muscle-bound, invincible macho man like [[Arnold Schwarzenegger]], who was offered the role, but he wanted to branch out into comedy and turned it down to star in ''[[Twins (1988 film)|Twins]]'' (1988).&lt;ref name=Independent1/&gt; Willis was known mainly for his comedic role in the romantic comedy television series ''[[Moonlighting (TV series)|Moonlighting]]'', starring opposite [[Cybill Shepherd]]. He declined the role because of his



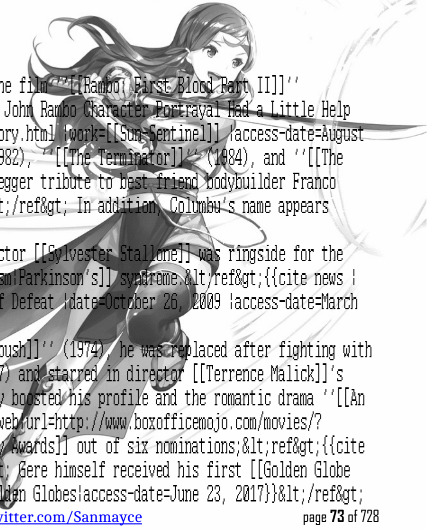
```
    contractual obligations to ''Moonlighting'', but when Shepherd became pregnant, the show's production was stopped for eleven weeks, giving Willis enough time to take the role.&lt;ref name=Independent1/&gt;
212 * [[Sylvester Stallone]] filmed scenes from the movie ''[[Cop Land]]'' in 1997 at the Hilltop Reservations.&lt;ref&gt;Kennedy, Richard A. [https://books.google.com/books?id=Gw47BAAAQBAJ&amp;pg=PA116 ''Essex Mountain Sanatorium''], p. 116. [[Arcadia Publishing]], 2013. {{ISBN|9781439643792}}. Accessed December 23, 2014. &quot;Other notable productions that used the property for filming were the Sylvester Stallone movie ''Copland'', Sonic Youth's video for their song 'Candle,' and various episodes of ''The Sopranos'',&quot;&lt;/ref&gt;
213 Some actors from both live-action and animated works have also reprised their respective roles in video games, such as [[Kevin Conroy]] ([[Batman]]) and [[Mark Hamill]] ([[The Joker]]) in the ''[[Batman: Arkham]]'' series, [[Sylvester Stallone]] ([[John Rambo]]) in ''[[Mortal Kombat 11]]'', various actors from the works of [[Walt Disney Animation Studios]] in ''[[Kingdom Hearts]]'', and [[Mike Pollock (voice actor)|Mike Pollock]] ([[Doctor Eggman]]) in ''[[Sonic the Hedgehog]]''.
214 [[Hal Ashby]] was next sent the script, who agreed to direct the film. Fonda was cast from the beginning as Sally Hyde, the housewife. A top box-office star was sought for the male lead to offset the grim nature of the story. [[Al Pacino]], [[Jack Nicholson]] and [[Sylvester Stallone]] were all offered the part, but declined.&lt;ref name=&quot;Devine1999&quot;&gt;{{cite book|last=Devine|first=Jeremy M.|title=Vietnam at 24 Frames a Second: A Critical and Thematic Analysis of Over 400 Films about the Vietnam War|url=https://books.google.com/books?id=RIgkkakqBYoC&amp;pg=PA148|year=1999|publisher=University of Texas Press|isbn=978-0-292-71601-8|page=148}}&lt;/ref&gt; [[Jon Voight]] had been considered for the role of the husband, but after becoming involved with the film, he campaigned to play the paraplegic veteran. Voight had participated in the anti-war movement and was a friend of Fonda, who was instrumental in helping him land the role, even though he had fallen from popularity since his ''Midnight Cowboy'' heyday. Bruce Dern, long stereotyped in sadistic roles, was chosen as the husband. The screenplay was written and rewritten until the project could wait no longer. Jane Fonda, who just finished ''[[Julia (1977 film)|Julia]]'' (1977), was soon to star in [[Alan J. Pakula]]'s ''[[Comes a Horseman]]'' (1978). For director Ashby, this was his second film about the 1960s, in addition to his 1975 film ''[[Shampoo (film)|Shampoo]]''.
215 In the late 1990s, it was reported that Oz was going to direct either [[Sylvester Stallone]] or [[Bruce Willis]] in an unmade film titled ''Ump'' for [[Metro-Goldwyn-Mayer]].&lt;ref&gt;{{cite magazine|last=Fleming|first=Michael|title=Stallone pumped about 'Ump'|date=6 January 1997|magazine=[[Variety (magazine)|Variety]]|url=https://variety.com/1997/voices/columns/stallone-pumped-about-ump-1117433935/amp/|access-date=1 July 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine|last2=Fleming|first2=Michael|last1=Carver|first1=Benedict|title=Willis may swing at 'Ump'|date=25 June 1998|magazine=[[Variety (magazine)|Variety]]|url=https://variety.com/1998/film/news/willis-may-swing-at-ump-1117477885/amp/|access-date=1 July 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite news|title=Bruce Willis May Be Out, But Movie 'Ump' Still On |date=24 December 1999|newspaper=[[Orlando Sentinel]]|url=https://www.orlandosentinel.com/news/os-xpm-1999-12-24-9912230426-story,amp.html|access-date=1 July 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite news|last=Campbell|first=Ramsey|title=MGM Studio Still Pitching Idea of Filming 'Ump' |date=23 August 2001|newspaper=[[Orlando Sentinel]]|url=https://www.orlandosentinel.com/news/os-xpm-2001-08-23-0108230592-story,amp.html|access-date=1 July 2019}}&lt;/ref&gt;
216 While some developers continued to focus on demolition, several investors like [[Tony Goldman]] and [[Ian Schrager]] bought art deco hotels and transformed them into world famous hot spots in the '80s and '90s. Among the celebrities that frequented Miami Beach were [[Madonna]], [[Sylvester Stallone]], [[Cher]], [[Oprah Winfrey]] and [[Gianni Versace]].&lt;ref&gt;{{Cite news |title=In two decades, Miami Beach's southernmost square mile has become a big draw with art, cuisine, architecture and nightlife. |last=Martin |first=Lydia |date=2007-07-15 |work=The Miami Herald |page=A1}}&lt;/ref&gt;
217 As Moe's voice is based on Al Pacino's, likewise many of Azaria's other recurring characters are based on existing sources. He took Apu's voice from the many Indian and Pakistani convenience store workers in Los Angeles that he had interacted with when he first moved to the area, and also loosely based it on [[Peter Sellers]]' character Hrundi V. Bakshi from the film ''[[The Party (1968 film)|The Party]]''.&lt;ref name=&quot;freshair&quot; /&gt; Originally, it was thought that Apu being Indian was too offensive and stereotyped, but after Azaria's reading of the line &quot;Hello, Mr. Homer&quot;, which the show's producers thought was hilarious, the character stayed.&lt;ref name=&quot;msn&quot; /&gt;&lt;ref name=&quot;tvguide&quot;&gt;{{cite news |title=Flash! 24 Simpsons Stars Reveal Themselves |url=http://simpsonsarchive.com/other/articles/flash.html |date=October 21, 2000 |work=[[TV Guide]] |first=Joe |last=Rhodes}}&lt;/ref&gt; Azaria, however, disputed this on ''LateNet with [[Ray Ellin]]'', claiming that Apu was always intended to be stereotypical.&lt;ref name=&quot;ellin&quot;&gt;{{cite interview|subject=Azaria, Hank |interviewer=[[Ray Ellin]] |title=Hank Azaria on LateNet with Ray Ellin |url=http://www.dailycomedy.com/joke/10559 |work=LateNet with Ray Ellin |publisher=Daily Comedy |date=December 5, 2007 |access-date=April 3, 2008 |url-status=dead |archive-url=https://web.archive.org/web/20080409175439/http://www.dailycomedy.com/joke/10559 |archive-date=April 9, 2008 }}&lt;/ref&gt; Chief Wiggum's voice was originally a parody of [[David Brinkley]], but when Azaria was told it was too slow, he switched it to that of [[Edward G. Robinson]].&lt;ref name=&quot;tvguide&quot; /&gt; Officer Lou is based on [[Sylvester Stallone]],&lt;ref name=&quot;barber&quot; /&gt; and Dr. Nick is &quot;a bad [[Ricky Ricardo]] impression.&quot;&lt;ref&gt;{{cite video |people=Azaria, Hank |date=2004 |title=The Simpsons The Complete Fourth Season DVD commentary for the episode &quot;[[Homer's Triple Bypass]]&quot; |medium=DVD |publisher=20th Century Fox}}&lt;/ref&gt; The &quot;[[List of recurring The Simpsons characters#Wiseguy|Wise Guy]]&quot; voice is &quot;basically [[Charles Bronson]],&quot;&lt;ref name=&quot;barber&quot;&gt;{{cite video |people=Azaria, Hank |date=2004 |title=The Simpsons The Complete Fifth Season DVD commentary for the episode &quot;[[Homer's Barbershop Quartet]]&quot; |medium=DVD |publisher=20th Century Fox}}&lt;/ref&gt; while Carl is &quot;a silly voice [Azaria] always did.&quot;&lt;ref name=&quot;enemy&quot; /&gt; Two of the voices come from his time at college: Snake's is based on Azaria's old college roommate, while Comic Book Guy's voice is based on a student who lived in the room next door to Azaria and went by the name &quot;F&quot;.&lt;ref name=&quot;barber&quot; /&gt; Professor Frink is based on [[Jerry Lewis]]' performance in the original ''[[The Nutty Professor (1963 film)|The Nutty Professor]]'', and the Sea Captain is based on English actor [[Robert Newton]]'s portrayal of many pirates.&lt;ref name=&quot;tvguide&quot; /&gt; Azaria based his performance for the one-time character [[Frank Grimes (Simpsons Character)|Frank Grimes]], from the episode &quot;[[Homer's Enemy]]&quot;, on actor [[William H. Macy]]. He counts Grimes as the hardest, most emotional performance he has ever had to give in the history of ''The Simpsons''.&lt;ref name=&quot;enemy&quot;&gt;{{cite video |people=Azaria, Hank |date=2006 |title=The Simpsons The Complete Eighth Season DVD commentary for the episode &quot;[[Homer's Enemy]]&quot; |medium=DVD |publisher=20th Century Fox}}&lt;/ref&gt;
218       <text bytes="36498" sha1="9od9uxpf4107m1q0nd9hqq030qtkasu" xml:space="preserve">{{short description|1982 film by Sylvester Stallone}}
219 | director       = [[Sylvester Stallone]]
220 | writer         = Sylvester Stallone
221 * Sylvester Stallone
222 '''''Rocky III''''' is a 1982 American [[Sports film|sports]] [[Drama (film and television)|drama film]] written, directed by, and starring [[Sylvester Stallone]].&lt;ref name=&quot;Rocky III&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/20807/Rocky-III/|title=Rocky III|work=TCM database|publisher=[[Turner Classic Movies]]|access-date=February 28, 2016|archive-date=March 4, 2016|archive-url=https://web.archive.org/web/20160304193146/http://www.tcm.com/tcmdb/title/20807/Rocky-III/|url-status=live}}&lt;/ref&gt; The film is the sequel to ''[[Rocky II]]'' (1979) and the third installment in the [[Rocky (franchise)|''Rocky'' film series]]. It also stars [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], and [[Burgess Meredith]]. In the film, [[Rocky Balboa]] (Stallone) faces stiff competition from [[Clubber Lang]] (Mr. T), a powerful new contender, and turns to his old adversary [[Apollo Creed]] (Weathers) to help him train.
223 * [[Sylvester Stallone]] as [[Rocky Balboa|Robert &quot;Rocky&quot; Balboa]], &quot;The Italian Stallion&quot;: the heavyweight champion of the world who continues to defend his title against other fighters. When Lang challenges Rocky and wins, the public cries for a rematch. As Rocky is reluctant following Mickey's death, former rival, Apollo Creed, befriends and trains The Italian Stallion in his preparation to take on Clubber Lang.
224 For the role of Clubber Lang, two real world-class heavyweight boxers were first considered: [[Joe Frazier]] and [[Earnie Shavers]]. Both were about the same height as Sylvester Stallone and had the powerful physique he was looking for, but, according to casting director Rhonda Young, Frazier had a stuttering problem, while Shavers had a high-pitched voice which would have undermined the character's menacing presence. After looking far and wide (even going to prisons in the hope of finding a suitable antagonist), Rhonda Young stumbled upon a television program on NBC, ''[[Games People Play (1980 TV series)|America's Toughest Bouncer]]'', showcasing a sports competition, and was mesmerized by the winner's physical prowess as well as his charisma. She then called the producer, Don Ohlmeyer, to ask him about &quot;the man with the mohawk&quot;. It turned out that not only was Mr. T a perfect fit, but he also proved tremendously determined to give the best possible performance right from the first [[screen test]], for what would be his [[breakout role]].&lt;ref&gt;{{Cite web|url=https://www.youtube.com/watch?v=m8uSM2YQoTA|title=Mr. T biography (1999) [starting from 08:40]|website=[[YouTube]]|access-date=2020-08-25|archive-date=2020-11-28|archive-url=https://web.archive.org/web/20201128193351/https://www.youtube.com/watch?v=m8uSM2YQoTA|url-status=live}}&lt;/ref&gt;
225 [[Gene Siskel]] gave the film two-and-a-half stars out of four and wrote, &quot;Sorry to say this, but there's not anything new in ''Rocky III'', and we sit there wondering why it exists.&quot; He added that &quot;we see nothing new about Rocky's character, except that the tender side of his soul, which made him so appealing, is now virtually missing. Rocky Balboa in ''Rocky III'' is no longer likable.&quot;&lt;ref&gt;Siskel, Gene (May 28, 1982). &quot;'Rocky' loses its punch in Round 3&quot;. ''[[Chicago Tribune]]''. Section 3, p. 3.&lt;/ref&gt; [[Pauline Kael]] of ''[[The New Yorker]]'' stated, &quot;The first ''Rocky'' was primitive in a relatively innocent way. This picture is primitive, but it's also shrewd and empty and inept.&quot;&lt;ref&gt;Kael, Pauline (May 31, 1982). &quot;The Current Cinema&quot;. ''[[The New Yorker]]''. 84.&lt;/ref&gt; [[Sheila Benson]] of the ''[[Los Angeles Times]]'' wrote, &quot;Somehow, Sylvester Stallone has kicked life into what you might imagine is a pretty tired Rocky Balboa and has gotten him up on his feet again ... ''Rocky III'' works, possibly even better than numbers ''I'' and ''II''.&quot;&lt;ref&gt;Benson, Sheila (May 28, 1982). &quot;Stallone's Punchout No. 3 at the K.O. Corral&quot;. ''[[Los Angeles Times]]''. Part VI, p. 1.&lt;/ref&gt; Rita Kempley of ''[[The Washington Post]]'' called it &quot;as much fun as ever, a ground-meat-and-potatoes movie, with guys beating hell out of each other to a disco beat.&quot;&lt;ref&gt;Kempley, Rita (May 28, 1982). &quot;'Rocky III': Punched Out&quot;. ''[[The Washington Post]]''. Weekend, p. 13.&lt;/ref&gt;&lt;ref&gt;{{cite news |date=May 28, 1982 |last=Kempley |first=Rita |author-link=Rita Kempley |title='Rocky III': Punched Out |url=https://www.washingtonpost.com/archive/lifestyle/1982/05/28/rocky-iii-punched-out/82174b27-f0fa-4ea7-9ceb-428d10e35e5c/ |newspaper=[[Washington Post]] |access-date=August 12, 2020 |archive-date=May 1, 2021 |archive-url=https://web.archive.org/web/20210501055755/https://www.washingtonpost.com/archive/lifestyle/1982/05/28/rocky-iii-punched-out/82174b27-f0fa-4ea7-9ceb-428d10e35e5c/ |url-status=live }}&lt;/ref&gt; [[Tom Milne]] of ''[[The Monthly Film Bulletin]]'' wrote, &quot;Starting off with a replay of our hero's second miraculous return from the dead to win the championship back at the end of ''Rocky II''-itself a virtual repeat from the original ''Rocky''-''Rocky III'' soon demonstrates that it has nothing to offer but more of the same ... There are fleeting moments, thanks chiefly to a personable performance from Carl Weathers, but the time has surely come for Rocky Balboa to take the final count.&quot;&lt;ref&gt;{{cite journal |last=Milne |first=Tom |date=August 1982 |title=Rocky III |journal=[[The Monthly Film Bulletin]] |volume=49 |issue=583 |page=174 }}&lt;/ref&gt;
226 | [[Sylvester Stallone]]
227 A bronze statue of Rocky, called &quot;ROCKY&quot;, was commissioned by Sylvester Stallone and created by [[A. Thomas Schomberg]] in 1981.&lt;ref&gt;{{Cite web|url=http://www.pophistorydig.com/topics/tag/rocky-statue-philadelphia-art-museum/|title=Rocky statue &amp; Philadelphia Art Museum {{!}} The Pop History Dig|language=en-US|access-date=2019-01-08|archive-date=2019-04-26|archive-url=https://web.archive.org/web/20190426060526/https://www.pophistorydig.com/topics/tag/rocky-statue-philadelphia-art-museum/|url-status=live}}&lt;/ref&gt; Three statues were created, and one was placed on the top of the steps of the [[Philadelphia Museum of Art]] for the filming of ''Rocky III''. After filming was complete, a furious debate erupted in Philadelphia between the Art Museum and the City's Art Commission over the meaning of &quot;[[art]]&quot;. Claiming the statue was not &quot;art&quot;, but rather a &quot;movie [[Theatrical property|prop]]&quot; the city considered various alternative locations and settled upon the front of the [[Spectrum (arena)|Spectrum]] in South Philadelphia. It was later returned to the Art Museum where it was used in the filming of ''[[Rocky V]]'',&lt;ref&gt;{{cite news |last=Hinds |first=Michael Decourcy |title=Love Gets in the Way as Rocky V Starts Filming
```



```
|url=https://www.nytimes.com/1990/01/23/movies/love-gets-in-the-way-as-rocky-v-starts-filming.html |access-date=21 September 2020 |work=[[The New York Times]] |date=January 23, 1990 |archive-date=1 October 2020 |archive-url=https://web.archive.org/web/20201001235608/https://www.nytimes.com/1990/01/23/movies/love-gets-in-the-way-as-rocky-v-starts-filming.html |url-status=live }}&lt;/ref&gt; as well as ''[[Mannequin (1987 film)|Mannequin]]'' and ''[[Philadelphia (film)|Philadelphia]]''. Afterward, it was again moved to the front of the Spectrum. The statue was returned to the bottom of the museum's stairs on 8 September 2006.&lt;ref&gt;{{Cite news|url=https://www.washingtonpost.com/wp-dyn/content/article/2006/09/08/AR2006090801620_pf.html?noredirect=on|title=Rocky Is Back Where He Belongs|newspaper=[[The Washington Post]]|access-date=2019-01-08|language=en-US|issn=0190-8286|archive-date=2019-01-09|archive-url=https://web.archive.org/web/20190109062838/http://www.washingtonpost.com/wp-dyn/content/article/2006/09/08/AR2006090801620_pf.html?noredirect=on|url-status=live}}&lt;/ref&gt; The steps leading to the east entrance of the Philadelphia Museum of Art are also known as "[[Rocky Steps|The Rocky Steps]]."&lt;ref&gt;{{Cite web|url=https://www.visitphilly.com/things-to-do/attractions/the-rocky-statue-and-the-rocky-steps/|title=Visit The Rocky Statue And The Rocky Steps|website=Visit Philadelphia|language=en-US|access-date=2019-01-08|archive-date=2019-01-08|archive-url=https://web.archive.org/web/20190108194149/https://www.visitphilly.com/things-to-do/attractions/the-rocky-statue-and-the-rocky-steps/|url-status=live}}&lt;/ref&gt; A [[Rocky statue in Žitište|similar statue]] is located in [[Žitište]], [[Serbia]].
228 {{Sylvester Stallone}}
229 [[Category:Films directed by Sylvester Stallone]]
230 [[Category:Films with screenplays by Sylvester Stallone]]
231 In ''[[Rocky IV]]'', the climactic fight between American [[Rocky Balboa]] and Russian [[Ivan Drago]] was inspired by the bout between [[Joe Louis]] and Max Schmeling, updated to reflect Cold War relationships.&lt;ref&gt;{{Citation |title=The Making of ROCKY VS. DRAGO by Sylvester Stallone | date=30 October 2021 |url=https://www.youtube.com/watch?v=suhqtn_brBs |access-date=2023-08-05 | language=en}}&lt;/ref&gt;
232 [[Ben Affleck and Matt Damon|Affleck and Damon]] proposed to act in the lead roles, but many studio executives said that they wanted [[Brad Pitt]] and [[Leonardo DiCaprio]].&lt;ref&gt;{{cite web | last=Juul |first=Matt |date=January 6, 2016 |title='Good Will Hunting' Might've Been Very Different Without Sylvester Stallone |url=https://www.bostonmagazine.com/arts-entertainment/2016/01/06/matt-damon-sylvester-stallone-rocky/ |url-status=live |archive-url=https://web.archive.org/web/20180215143637/https://www.bostonmagazine.com/arts-entertainment/2016/01/06/matt-damon-sylvester-stallone-rocky/ |archive-date=February 15, 2018 |access-date=March 11, 2018 |work=[[Boston Magazine]]}}&lt;/ref&gt; Meanwhile, [[Kevin Smith]] was working with Affleck on ''[[Mallrats]]'' and with both Damon and Affleck on ''[[Chasing Amy]].''&lt;ref&gt;Smith's comments on the ''[[Mallrats]]'' DVD audio commentary&lt;/ref&gt; Castle Rock Entertainment put the script in turnaround and gave Damon and Affleck 30 days to find another buyer for the script who would reimburse Castle Rock Entertainment the money paid; otherwise the script would be reverted to Castle Rock Entertainment, and Damon and Affleck would be out. All of the studios that were involved in the original bidding war for the screenplay turned the pair down, taking meetings with Affleck and Damon only to tell them this to their face.
233 At Pinewood Studios northwest of London, and on location in [[Socialist Federal Republic of Yugoslavia|Yugoslavia]], he worked on the musical ''[[Fiddler on the Roof (film)|Fiddler on the Roof]]'' (1971, re-issued 1979), which won three [[Academy Awards|Oscars]] and was nominated for five others, including Best Picture and Director. During the filming of ''Fiddler'', Jewison was also the subject of the 1971 [[National Film Board of Canada]] documentary, ''Norman Jewison, Filmmaker'', directed by Douglas Jackson.&lt;ref&gt;{{cite web|url=http://www.onf-nfb.gc.ca/eng/collection/film/?id=10595 |title=National Film Board of Canada archives &quot;Norman Jewison, Filmmaker&quot; |publisher=Onf-nfb.gc.ca |date=August 3, 2010 |access-date=June 4, 2011}}&lt;/ref&gt; Jewison's next project was the musical ''[[Jesus Christ Superstar (film)|Jesus Christ Superstar]]'' (1973), based on the Broadway musical written by [[Andrew Lloyd Webber]] and [[Tim Rice]]. It was filmed in Israel, where Jewison also produced the western ''Billy Two Hats'' (1974), starring [[Gregory Peck]]. ''Superstar'', controversial for its treatment of a religious subject, was followed by another movie that sparked critical debate, this time over violence. ''[[Rollerball (1975 film)|Rollerball]]'' (1975) is set in the near future when corporations rule the world and entertainment is centred around a deadly game. The next film he directed, the labour union drama ''[[F.I.S.T. (film)|F.I.S.T.]]'' (1978), loosely based on the life of [[Jimmy Hoffa]], also provided some controversy, this time regarding the [[screenplay|screenwriting]] credit. Screenwriter [[Joe Eszterhas]] was unhappy to share the screenwriting credit with the film's star [[Sylvester Stallone]], as he felt that Stallone's input had been minor, while Stallone claimed to have basically rewritten the whole script.&lt;ref&gt;{{cite web|url=https://tv.yahoo.com/joe-eszterhas/contributor/30330/bio |title=Yahoo! TV Esterhaus biography |publisher=Tv.yahoo.com |access-date=June 4, 2011}}&lt;/ref&gt;
234 Sutherland received praise for his role as the conflicted and grieving father in the [[Robert Redford]] directed family drama ''[[Ordinary People]]'' (1980), alongside [[Mary Tyler Moore]] and [[Timothy Hutton]].&lt;ref name=&quot;TG&quot;&gt;&lt;/ref&gt;&lt;ref&gt;{{Cite web |access-date=22 June 2024 |title=Donald Sutherland dies: Five essential roles by the Canadian acting legend |url=https://www.thealbertan.com/national-entertainment/donald-sutherland-dies-five-essential-roles-by-the-canadian-acting-legend-9113964 |date=21 June 2024 |publisher=[[The Albertan]] |archive-date=22 June 2024 |archive-url=https://web.archive.org/web/20240622030716/https://www.thealbertan.com/national-entertainment/donald-sutherland-dies-five-essential-roles-by-the-canadian-acting-legend-9113964 |url-status=live }}&lt;/ref&gt; In September 1980, [[Vincent Canby]] of ''[[The New York Times]]'' wrote, &quot;Mr. Sutherland realizes his best film role in years, playing a fellow who, filled with love for both his wife and his son, is angrily accused by each of fence-sitting, of being weak and indecisive when he's really the only one in the family with some idea of what is wrong.&quot;&lt;ref&gt;{{cite news|first=Vincnt|last=Canby|author-link=Vincent Canby|date=September 19, 1980|url= https://archive.nytimes.com/www.nytimes.com/packages/html/movies/bestpictures/people-re.html|title= Review: Redford's Ordinary People|work=[[The New York Times]]|accessdate=June 22, 2024}}&lt;/ref&gt; Sutherland was nominated for the [[Golden Globe Award for Best Actor in a Motion Picture - Drama]].&lt;ref&gt;{{cite news|url=https://www.nytimes.com/1981/01/14/movies/ordinary-people-tops-golden-globe-nominations.html|title=Ordinary People tops 10 Golden Globe Award Nominations|work=[[The New York Times]]|agency=[[United Press International]]|date=January 14, 1981|accessdate=June 22, 2024}}&lt;/ref&gt; In 1981, he starred in the English [[spy thriller]] ''[[Eye of the Needle (film)|Eye of the Needle]]''&lt;ref&gt;{{Cite web |last=Ebert|first=Roger|author-link=Roger Ebert|title=Eye of the Needle movie review (1981)|url=https://www.rogerebert.com/reviews/eye-of-the-needle-1981 |access-date=2024-06-22 |website=[[RogerEbert.com]] |language=en |date=January 1, 1981 |archive-date=28 May 2023 |archive-url=https://web.archive.org/web/20230528193006/https://www.rogerebert.com/reviews/eye-of-the-needle-1981 |url-status=live }}&lt;/ref&gt; and narrated [[Anne Wheeler]]'s Canadian [[docudrama]] ''[[A War Story]]''.&lt;ref&gt;{{Cite web|title=A War Story|url=https://www.nfb.ca/film/war_story/|website=[[National Film Board of Canada]] |access-date=2024-06-22|language=en | archive-date=24 February 2024 |archive-url=https://web.archive.org/web/20240224223451/https://www.nfb.ca/film/war_story/ |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Ohayon | first=Albert |date=2016-11-09 |title=A War Story: Triumph of the Human Spirit |url=https://blog.nfb.ca/blog/2016/11/09/war-story-triumph-human-spirit/ |access-date=2024-06-22 |website=NFB Blog | language=en-CA |archive-date=10 December 2023 |archive-url=https://web.archive.org/web/20231210134859/https://blog.nfb.ca/blog/2016/11/09/war-story-triumph-human-spirit/ |url-status=live }}&lt;/ref&gt; He played the role of physician-hero [[Norman Bethune]] in ''Bethune'' (1977) and ''[[Bethune: The Making of a Hero]]'' (1990).&lt;ref name=&quot;CBCBethune&quot;/&gt;&lt;ref&gt;{{Cite news|first=Brian D.|last=Johnson|author-link=Brian D. Johnson|date=August 10, 1987|title=Making A Legend |url=https://archive.macleans.ca/article/1987/8/10/making-a-legend|work=[[Maclean's]] |access-date=2024-06-22 |archive-date=19 December 2022 |archive-url=https://web.archive.org/web/20221219162453/https://archive.macleans.ca/article/1987/8/10/making-a-legend |url-status=bot: unknown }}&lt;/ref&gt; In 1983, he co-starred with [[Teri Garr]] and [[Tuesday Weld]] in an adaptation of [[John Steinbeck]]'s ''[[The Winter of Our Discontent (film)|The Winter of Our Discontent]]''.&lt;ref&gt;{{Cite news |last=O'Connor |first=John J. |date=1983-12-06 |title=TV: ADAPTED STEINBECK |url=https://www.nytimes.com/1983/12/06/arts/tv-adapted-steinbeck.html |access-date=2024-06-22 |work=The New York Times |language=en-US |issn=0362-4331 |archive-date=24 May 2015 |archive-url=https://web.archive.org/web/20150524134928/http://www.nytimes.com/1983/12/06/arts/tv-adapted-steinbeck.html |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Hill |first=Michael E. |date=December 3, 1983 |title='Winter of Our Discontent'/Putting Steinbeck on the Screen | newspaper=[[The Washington Post]] |url=https://www.washingtonpost.com/archive/lifestyle/tv/1983/12/04/winter-of-our-discontent-putting-steinbeck-on-the-screen/28b9375c-3b32-4d50-baee-2f732e60a511/ | url-status=live |access-date=22 June 2024 |archive-date=28 August 2017 |archive-url=https://web.archive.org/web/20170828110637/https://www.washingtonpost.com/archive/lifestyle/tv/1983/12/04/winter-of-our-discontent-putting-steinbeck-on-the-screen/28b9375c-3b32-4d50-baee-2f732e60a511/ }}&lt;/ref&gt; Some of Sutherland's better known roles in the 1980s and 1990s were in the apartheid drama ''[[A Dry White Season]]'' (1989), alongside [[Marlon Brando]] and [[Susan Sarandon]];&lt;ref&gt;{{Cite web |date=2012-10-12 |title=A Dry White Season :: rogerebert.com :: Reviews |url=http://rogerebert.suntimes.com/apps/pbcs.dll/article?AID=/19890922/REVIEWS/909220302 |access-date=2024-06-22 |archive-date=12 October 2012 |archive-url=https://web.archive.org/web/20121012074105/http://rogerebert.suntimes.com/apps/pbcs.dll/article?AID=/19890922/REVIEWS/909220302 |url-status=dead }}&lt;/ref&gt; as a sadistic warden in ''[[Lock Up (1989 film)|Lock Up]]'' (1989) with [[Sylvester Stallone]];&lt;ref&gt;{{Cite web |last=Thomas |first=Kevin |date=1989-08-04 |title=MOVIE REVIEW : More Muscular Madness From Stallone in 'Lock Up' |url=https://www.latimes.com/archives/la-xpm-1989-08-04-ca-540-story.html |access-date=2024-06-22 |website=Los Angeles Times |language=en-US |archive-date=3 September 2023 |archive-url=https://web.archive.org/web/20230903071553/https://www.latimes.com/archives/la-xpm-1989-08-04-ca-540-story.html |url-status=live }}&lt;/ref&gt; as an incarcerated pyromaniac in the [[firefighter]] thriller ''[[Backdraft (film)|Backdraft]]'' (1991) alongside [[Kurt Russell]] and [[Robert De Niro]],&lt;ref&gt;{{Cite magazine |date=2010-01-10 |title=Backdraft {{!}} Movies | magazine=[[Entertainment Weekly]] |url=http://www.ew.com/ew/article/0,,314445,00.html |access-date=2024-06-22 |archive-date=10 January 2010 |archive-url=https://web.archive.org/web/20100110001150/http://www.ew.com/ew/article/0,,314445,00.html |url-status=bot: unknown }}&lt;/ref&gt; as the humanitarian doctor-activist [[Norman Bethune]] in 1990's ''[[Bethune: The Making of a Hero]]'',&lt;ref&gt;{{Cite news |date=1990-08-28 |title=Bethune saga finally hits the big screen |url=https://www.newspapers.com/article/the-gazette/114860808/ |access-date=2024-06-22 |work=[[The Gazette (Montreal)|The Gazette]] |pages=40 |archive-date=22 June 2024 |archive-url=https://web.archive.org/web/20240622045749/https://www.newspapers.com/article/the-gazette/114860808/ |url-status=live }}&lt;/ref&gt; and as a snobbish New York City [[art dealer]] in ''[[Six Degrees of Separation (film)|Six Degrees of Separation]]'' (1993), with [[Stockard Channing]] and [[Will Smith]].&lt;ref&gt;{{Cite web |date=2024-06-21 |title=Will Smith Pays Tribute to Late &quot;Six Degrees of Separation&quot; Costar Donald Sutherland: 'Rest in Peace, Donald' | url=https://www.yahoo.com/entertainment/smith-pays-tribute-six-degrees-032119196.html |access-date=2024-06-22 |website=Yahoo Entertainment |language=en-US |archive-date=22 June 2024 |archive-url=https://web.archive.org/web/20240622045749/https://www.yahoo.com/entertainment/smith-pays-tribute-six-degrees-032119196.html |url-status=live }}&lt;/ref&gt;
235 * ''[[Tulsa King]]'' TV series starring [[Sylvester Stallone]] (2022)
236 Chan rekindled his [[Cinema of the United States|Hollywood]] ambitions in the 1990s, but refused early offers to play villains in Hollywood films to avoid being [[typecasting|typecast]] in future roles. For example, [[Sylvester Stallone]] offered him the role of [[Simon Phoenix]], a criminal in the futuristic film ''[[Demolition Man (film)|Demolition Man]]''. Chan declined and the role was taken by [[Wesley Snipes]].&lt;ref&gt;{{cite web |last=Dickerson |first=Jeff |title=Black Delights in Demolition Man |work=The Michigan Daily |date=4 April 2002 |url=http://media.www.michigandaily.com/media/storage/paper851/news/2002/04/04/TheStatement/Black.Delights.In.demolition.Man-1403498.shtml |archive-url=https://web.archive.org/web/20071224001408/http://media.www.michigandaily.com/media/storage/paper851/news/2002/04/04/TheStatement/Black.Delights.In.demolition.Man-1403498.shtml |archive-date=24 December 2007 |access-date=29 February 2012}}&lt;/ref&gt;
237 American filmmaker [[Quentin Tarantino]] classified Chan's style of acting and filmmaking as [[physical comedy]], and considered him one of the greatest in the genre.&lt;ref name=&quot;Utah&quot; /&gt; British filmmaker [[Edgar Wright]] describes Jackie Chan as an &quot;expressive&quot; visual performer with an [[everyman]] persona. He notes that, in contrast to other action heroes (such as Bruce Lee, [[Sylvester Stallone]], [[Clint Eastwood]] or [[Arnold Schwarzenegger]]), Chan presents himself as a loveable &quot;goofball&quot; underdog who overcomes the odds with almost &quot;superhuman&quot; acrobatic stunts and fighting abilities.&lt;ref name=&quot;Wright&quot;&gt;{{cite news |last1=Russell |first1=Calum |title=Edgar Wright compares the flair of Jackie Chan and Bruce Lee |url=https://faroutmagazine.co.uk/edgar-wright-compares-the-genius-of-jackie-chan-and-bruce-lee/ |access-date=17 March 2022 |work=[[Far Out Magazine]] |date=26 November 2021}}&lt;/ref&gt;
238 * [[Don Kernodle]], born in Burlington, five-time NWA champion and tag team partner of Sgt. Slaughter; appeared in ''[[Paradise Alley]]'' with [[Sylvester Stallone]]
```



```
239 * '''''[[Action films|Action]]''''': In the 1970s, action films usually focused on maverick police officers. However, the genre did not become dominant in Hollywood until the 1980s, when it was popularized by action stars such as [[Arnold Schwarzenegger]], [[Sylvester Stallone]], [[Chuck Norris]], and [[Bruce Willis]]. Schwarzenegger built an iconic career out of action classics such as ''[[The Terminator]]'' (1984), ''[[Commando (1985 film)|Commando]]'' (1985), and ''[[Predator (film)|Predator]]'' (1987). Stallone starred in 1982's ''[[First Blood]]'' about a returning [[Vietnam War]] veteran fighting a small town sheriff, as well as its sequels ''[[Rambo: First Blood Part II]]'' and ''[[Rambo III]]''. Previously seen as a taboo in the 1970s, Vietnam War flicks like [[Oliver Stone]]'s ''[[Platoon (film)|Platoon]]'' and [[Stanley Kubrick]]'s ''[[Full Metal Jacket]]'' made the conflict a popular subject in the 1980s. Norris starred in the ''[[Missing in Action (film)|Missing in Action]]'' trilogy (1984, 1985, 1988) about a Vietnam veteran going back to rescue [[POW]]s. 1988's ''[[Die Hard]]'' was particularly influential on the development of the genre in the subsequent decade. In the film, Willis plays a New York City police detective who inadvertently becomes embroiled in a terrorist take-over of a [[Los Angeles, California|Los Angeles]] office building. Meanwhile, [[Hong Kong action cinema]] was being revolutionized by filmmakers [[Jackie Chan]], [[Tsui Hark]], and [[John Woo]]; garnering increased attention all over the world with the likes of ''[[Project A (film)|Project A]]'' (1983), ''[[Police Story (1985 film)|Police Story]]'' (1985), ''[[A Better Tomorrow]]'' (1986), and ''[[The Killer (1989 film)|The Killer]]'' (1989). Which featured increasingly complex [[Combat in film|martial arts and gunfight choreography]] with generally unsafe and most often uninsured [[stunt]] work. Blazing the trail for newer types of action movies like the [[heroic bloodshed]] subgenre. ''[[Tango &amp; Cash]]'', starring Stallone and [[Kurt Russell]], was one of the two last Hollywood films to be released in the 1980s.&lt;ref name=numbers&gt;{{cite news|title= Tango &amp; Cash - Box Office Data |work= The Numbers|url=http://www.the-numbers.com/movies/1989/0TNGC.php|access-date=2011-07-24}}&lt;/ref&gt;
240 [[East Jersey State Prison]], formerly known as Rahway State Prison, actually is located in [[Woodbridge Township, New Jersey|Woodbridge Township]] at the border with Rahway. The prison's mailing address is in Rahway, leading many to believe the facility was located there. The prison's official name was changed to East Jersey State Prison as of November 30, 1988, at the request of the citizens of Rahway.&lt;ref&gt;Malwitz, Rick. [https://web.archive.org/web/20130723200059/http://pqasb.pqarchiver.com/mycentraljersey/access/1846003261.html?FMT=ABS&amp;FMTS=ABS:FT&amp;type=current&amp;date=Feb+15%2C+2001&amp;author=RICK+MALWITZ&amp;pub=Home+News+Tribune&amp;desc=What%27s+in+a+name%3F+Plenty+if+we%27re+talking+prison&amp;pqatl=google &quot;What's in a name? Plenty if we're talking prison&quot;], ''[[Home News Tribune]]'', February 15, 2001. Accessed May 20, 2012. &quot;The name was officially changed to East Jersey State Prison Nov. 30, 1988.&quot;&lt;/ref&gt; East Jersey State Prison is seen at the beginning of the movie ''[[Ocean's Eleven (2001 film)|Ocean's Eleven]]'', starring [[George Clooney]]. The 1978 documentary ''[[Scared Straight]]'' was filmed there, as was the 1989 movie ''[[Lock Up (1989 film)|Lock Up]]'', starring [[Sylvester Stallone]]. The prison was briefly mentioned in [[John Sayles]] ''[[City of Hope (1991 film)|City of Hope]]'' (1991).&lt;ref&gt;[http://www.injersey.com/day/story/0,2379,355117,00.html What's in a name? Plenty if we're talking prison], [[Home News Tribune]], February 15, 2001.&lt;/ref&gt;
241       <text bytes="23202" sha1="6zoyrfwarsrsb7ry5u6vmsz44dra0nk" xml:space="preserve">{{Short description|1979 film by Sylvester Stallone}}
242 | director       = [[Sylvester Stallone]]
243 | writer         = Sylvester Stallone
244 * Sylvester Stallone
245 '''''Rocky II''''' is a 1979 American [[Sports film|sports]] [[Drama (film and television)|drama film]] written, directed by, and starring [[Sylvester Stallone]].&lt;ref name=&quot;Rocky II&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/20654/Rocky-II/|title=Rocky II|work=TCM database|publisher=[[Turner Classic Movies]]|access-date=February 27, 2016}}&lt;/ref&gt; The film is the sequel to ''[[Rocky]]'' (1976) and the second installment in the [[Rocky (franchise)|''Rocky'' film series]]. It also stars [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], and [[Burgess Meredith]]. In the film, [[Rocky Balboa]] (Stallone), struggling to adjust to his newfound fame and family life, finds himself in a rematch fiercely demanded by [[Apollo Creed]] (Weathers).
246 * [[Sylvester Stallone]] as [[Rocky Balboa|Robert &quot;Rocky&quot; Balboa]], &quot;The Italian Stallion&quot;: the underdog who was given one chance at winning the heavyweight championship from Apollo Creed in the first film. Due to the public's belief that it was very possible that Rocky could have won, he gets a second shot at the title in this film.
247 The story development of Rocky surprising Creed by switching to fighting right-handed was not in the original script and only came about because of an accident on set. While getting in shape for the film, Stallone experienced an almost complete tear on his right pectoralis major muscle while trying to bench press 100&amp;nbsp;kg with bodybuilder [[Franco Columbu]] and underwent a partially successful surgery in order to try to reattach the muscle. Therefore, he could not fight with his left hand.&lt;ref&gt;{{cite news |url=https://www.nytimes.com/1979/06/13/archives/rocky-fights-it-over-sports-of-the-times.html|title=Rocky Fights It over|newspaper=The New York Times|date=13 June 1979}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://theweek.com/articles/491907/sylvester-stallones-aging-physique|title=Sylvester Stallone's aging physique}}&lt;/ref&gt;
248 A [[novelization]] was published by [[Ballantine Books]] in 1979. Sylvester Stallone was credited as the author. The book is a first-person narrative told by Rocky himself.&lt;ref&gt;{{cite book|title=Rocky II (Book, 1979) |publisher=[WorldCat.org] |date=2016-05-11 |oclc = 5280245}}&lt;/ref&gt;
249 {{Sylvester Stallone}}
250 [[Category:Films directed by Sylvester Stallone]]
251 [[Category:Films with screenplays by Sylvester Stallone]]
252 King arranged Ali's [[Muhammad Ali vs. Chuck Wepner|1975 fight against journeyman Chuck Wepner]].&lt;ref&gt;{{cite news|last1=Smith|first1=Red|title=But Wasn't It a Bleedin' Shame?|url=https://www.nytimes.com/1975/03/25/archives/but-wasnt-it-a-bleedin-shame.html?_r=0|newspaper=The New York Times|date=March 25, 1975|access-date=March 30, 2017}}&lt;/ref&gt; It is widely believed the fight inspired [[Sylvester Stallone]] to write the screenplay for ''[[Rocky]]'' (1976).&lt;ref&gt;{{cite web|last1=Child|first1=Ben|title='Real-life Rocky' to sue over copycat film based on heavyweight contender's life|url=https://www.theguardian.com/film/2016/feb/23/real-life-rocky-chuck-wepner-to-sue-over-copycat-heavyweight-biopic|website=theguardian|date=February 23, 2016|publisher=The Guardian|access-date=March 30, 2017}}&lt;/ref&gt;
253 | writer         = [[Sylvester Stallone]]
254 | based_on       = {{Based on|[[List of Rocky characters|Characters]]|Sylvester Stallone}}
255 * Sylvester Stallone
256 '''''Rocky V''''' is a 1990 American [[Sports film|sports]] [[Drama (film and television)|drama film]] directed by [[John G. Avildsen]] and written by and starring [[Sylvester Stallone]]. It is the sequel to ''[[Rocky IV]]'' (1985) and the fifth installment in the [[Rocky (franchise)|''Rocky'' film series]].&lt;ref name=&quot;Rocky V&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/21906/Rocky-V/|title=Rocky V|work=TCM database|publisher=[[Turner Classic Movies]]|access-date=February 28, 2016}}&lt;/ref&gt; It also stars [[Talia Shire]], [[Burt Young]], [[Sage Stallone]] in his film debut, [[Tommy Morrison]], and [[Burgess Meredith]]. In the film, a financially struggling [[Rocky Balboa]] (Stallone) agrees to train protégé Tommy Gunn (Morrison) at the gym once owned by Balboa's trainer, [[Mickey Goldmill]] (Meredith).&lt;ref&gt;{{cite news|title= Film Flam for 'Rocky' |work=[[The New York Times]] |date=November 15, 1989|url= https://www.nytimes.com/1989/11/15/sports/film-flam-for-rocky.html |access-date=August 12, 2010 | first=Phil | last=Berger}}&lt;/ref&gt;
257 * [[Sylvester Stallone]] as [[Rocky Balboa|Robert &quot;Rocky&quot; Balboa]], &quot;The Italian Stallion&quot;: Heavyweight Champion of the World, who suffers from brain damage caused by the head-trauma he received at the hands of [[Ivan Drago]] in the previous film. Because of his injuries, Rocky is forced to officially retire from boxing. After moving back to Philadelphia, Rocky trains and manages underdog boxer Tommy Gunn and helps him to rise to fame.
258 In the years following the film's release, Stallone acknowledged that the injury which forces Rocky to retire, referenced in the film as a potentially lethal form of '[[brain damage]]', was inaccurate.&lt;ref&gt;{{cite web|url=http://www.aintitcool.com/node/30861 |title=Round One With Sylvester Stallone Q&amp;A!! |website=Aintitcool.com |access-date=December 1, 2015}}&lt;/ref&gt; Stallone stated that having discussed the story with many boxing medical professionals, the injury Rocky suffered was a milder form of brain damage, similar to that of a long term concussion that many boxers suffer from and by modern-day standards are still able to gain licenses to box. It would not have prevented Rocky from gaining a license to box nor killed him.&lt;ref name=AICNQA&gt;{{cite web|author=Moriaty|url=http://www.aintitcool.com/node/30861|title=Round One With Sylvester Stallone Q&amp;A!!|website=[[Ain't It Cool News]]|date=December 1, 2006|access-date=June 15, 2009}}&lt;/ref&gt;
259 Upon its release, the ''Los Angeles Times'' regarded it as the best of the ''Rocky'' sequels.&lt;ref&gt;{{cite news|title= MOVIE REVIEW : A Kinder, Gentler Rocky Balboa : Of Sylvester Stallone's 'Rocky' sequels, No. 5 comes closest to some of the endearing qualities associated with the first |work=Los Angeles Times|date=November 16, 1990|url=https://www.latimes.com/archives/la-xpm-1990-11-16-ca-4530-story.html |access-date=October 29, 2010|first=Michael|last=Wilmington}}&lt;/ref&gt;
260 Sylvester Stallone has publicly expressed hatred towards ''Rocky V'', giving it a 0 out of 10.&lt;ref&gt;{{Citation |title=Sylvester Stallone - Friday Night with Jonathan Ross (FULL INTERVIEW) |date=September 10, 2012 |url=https://www.youtube.com/watch?v=8ueiOX1Zxc |language=en |access-date=2022-07-20}}&lt;/ref&gt;
261 {{Sylvester Stallone}}
262 [[Category:Films with screenplays by Sylvester Stallone]]
263 Columbu appeared as himself in the 1977 bodybuilding docudrama ''[[Pumping Iron]]''. He was the body building coach for [[Sylvester Stallone]] in the film ''[[Rambo: First Blood Part II]]'' (1985),&lt;ref&gt;{{cite news |last=Marbella |first=Jean |date=October 28, 1985 |title=Dr. Rambo Sylvester Stallone's Body-Beautiful Status for His John Rambo Character Portrayal Had a Little Help from the Real Man Behind the Muscle, Bodybuilder/Chiropractor Franco Columbu |url=https://www.sun-sentinel.com/news/fl-xpm-1985-10-28-8502170296-story.html |work=[[Sun-Sentinel]] |access-date=August 31, 2019}}&lt;/ref&gt; and had roles in films that starred Schwarzenegger, including ''[[Conan the Barbarian (1982 film)|Conan the Barbarian]]'' (1982), ''[[The Terminator]]'' (1984), and ''[[The Running Man (1987 film)|The Running Man]]'' (1987).&lt;ref&gt;{{cite news |last=Horswill |first=Ian |date=September 3, 2019 |title=Arnold Schwarzenegger tribute to best friend bodybuilder Franco Columbu |url=https://news.theceomagazine.com/lifestyle/schwarzenegger-tribute-franco-columbu/ |work=CEO Magazine |access-date=September 3, 2019}}&lt;/ref&gt; In addition, Columbu's name appears during the opening credits of Schwarzenegger's ''[[Last Action Hero]]'' as the director of fictional film ''Jack Slater IV''.
264 Holmes's trainer [[Richie Giachetti]] called Holmes's pummeling of Ali &quot;awful&amp;nbsp;... the worst sports event I ever had to cover.&quot; Actor [[Sylvester Stallone]] was ringside for the fight and said that it was like watching an autopsy on a man who is still alive. The Holmes fight is said to have contributed to Ali's [[Parkinsonism|Parkinson's]] syndrome.&lt;ref&gt;{{cite news |url=https://www.nytimes.com/2009/10/27/arts/television/27muhammad.html |work=The New York Times |title=Boxing King Casts His Shadow, Even at Time of Defeat |date=October 26, 2009 |access-date=March 5, 2012 |first1=Mike |last1=Hale}}&lt;/ref&gt;
265 Gere began appearing in [[Hollywood (film industry)|Hollywood]] films in the mid-1970s. Originally cast in a starring role in ''[[The Lords of Flatbush]]'' (1974), he was replaced after fighting with his co-star [[Sylvester Stallone]]. Gere played a small but significant part in ''[[Looking for Mr. Goodbar (film)|Looking for Mr. Goodbar]]'' (1977) and starred in director [[Terrence Malick]]'s well-reviewed drama ''[[Days of Heaven]]'' (1978).&lt;ref name=&quot;actors&quot; /&gt; The crime drama ''[[American Gigolo]]'' (1980) significantly boosted his profile and the romantic drama ''[[An Officer and a Gentleman]]'' (1982) (co-starring [[Debra Winger]]) cemented Gere's ascent to stardom, grossing almost $130 million&lt;ref&gt;{{cite web|url=http://www.boxofficemojo.com/movies/?id=officerandagentleman.htm|title=An Officer and a Gentleman|website=Box Office Mojo|access-date=May 4, 2009}}&lt;/ref&gt; and winning two [[Academy Awards]] out of six nominations;&lt;ref&gt;{{cite web|title=The 55th Academy Awards - 1983|url=https://www.oscars.org/oscars/ceremonies/1983|website=Oscars.org|access-date=June 23, 2017}}&lt;/ref&gt; Gere himself received his first [[Golden Globe Award]] nomination.&lt;ref&gt;{{cite news|title=Officer and a Gentleman, An|url=http://www.goldenglobes.com/film/officer-and-gentleman|newspaper=Golden Globes|access-date=June 23, 2017}}&lt;/ref&gt;
```



For the remainder of the 1980s, Gere appeared in films of varying critical and commercial reception.&lt;ref name=&quot;mojo&quot;&gt;{{cite web|title=Richard Gere|url=http://www.boxofficemojo.com/people/chart/?view=Actor&amp;id=richardgere.htm|website=Box Office Mojo|access-date=May 4, 2009}}&lt;/ref&gt;&lt;ref name=&quot;rotten&quot;&gt;{{cite web|url=http://www.rottentomatoes.com/celebrity/richard_gere/|title=Richard Gere|website=Rotten Tomatoes|access-date=May 4, 2009}}&lt;/ref&gt; His career rebounded with the releases of ''[[Internal Affairs (film)|Internal Affairs]]'' (1990) and ''[[Pretty Woman]]'' (1990), the latter of which earned Gere his second [[Golden Globe Award]] nomination.&lt;ref&gt;{{cite web|title=Richard Gere|url=http://www.goldenglobes.com/person/richard-gere|website=Goldenglobes.com|access-date=June 24, 2017}}&lt;/ref&gt; The 1990s saw Gere star in successful films including ''[[Sommersby]]'' (1993) (opposite [[Jodie Foster]]), ''[[Primal Fear (film)|Primal Fear]]'' (1996) and ''[[Runaway Bride (film)|Runaway Bride]]'' (1999) (which reunited him with his ''Pretty Woman'' co-star [[Julia Roberts]]).&lt;ref name=&quot;mojo&quot; /&gt; He also took a leading role in the action thriller ''[[The Jackal (1997 film)|The Jackal]]'' (1997), playing former IRA militant Declan Mulqueen; Gere affected an Irish accent for the role.&lt;ref&gt;{{cite web|url=http://metro.co.uk/2011/03/16/st-patricks-day-top-5-worst-irish-accents-in-films-644648/|title=Top 5 worst Irish accents in films|date=March 16, 2011|publisher=Metro|access-date=June 24, 2017}}&lt;/ref&gt;

266 The casting of the film was a rather lengthy process. Marshall had initially considered [[Christopher Reeve]], [[Daniel Day-Lewis]], [[Kevin Kline]], and [[Denzel Washington]] for the role of Edward, and [[Albert Brooks]],&lt;ref&gt;{{Cite web |date=June 30, 2011 |title=The Lost Roles of Albert Brooks |url=https://www.vulture.com/amp/2011/06/the-lost-roles-of-albert-brooks.html |url-status=live |archive-url=https://web.archive.org/web/20210704080548/https://www.vulture.com/amp/2011/06/the-lost-roles-of-albert-brooks.html |archive-date=July 4, 2021 |access-date=June 15, 2021 |website=[[Vulture (website)|Vulture]]}}&lt;/ref&gt; [[Sylvester Stallone]], [[Al Pacino]] and [[Burt Reynolds]] turned it down.&lt;ref&gt;{{cite web |title=Burt Reynolds Reflects on the Roles He's Turned Down |last=Schaffstall |first=Katherine |date=March 16, 2018 |url=https://www.hollywoodreporter.com/tv/tv-news/burt-reynolds-reflects-roles-hes-turned-down-1095043/ |work=The Hollywood Reporter |access-date=April 13, 2023}}&lt;/ref&gt;&lt;ref name=Collider&gt;{{Cite web|url=https://collider.com/galleries/pretty-woman-behind-the-scenes-facts/|title=Surprising Behind-the-Scenes Facts about Pretty Woman|first=Tom|last=Reimann|date=April 25, 2019|website=Collider|access-date=January 2, 2023|archive-date=January 2, 2023|archive-url=https://web.archive.org/web/20230102063114/https://collider.com/galleries/pretty-woman-behind-the-scenes-facts/|url-status=live}}&lt;/ref&gt; Pacino went as far as doing a casting reading with Roberts before rejecting the part.&lt;ref&gt;{{cite interview |first= Al |last= Pacino |subject-link= Al Pacino |date= June 15, 2007 |url= http://transcripts.cnn.com/TRANSCRIPTS/0706/15/lkl.01.html |title= ''&quot;Al Pacino Interview&quot;'' |interviewer= [[Larry King]] |work= [[Larry King Live]] |publisher= [[CNN]] |access-date= June 16, 2007 |archive-url= https://web.archive.org/web/20070621070302/http://transcripts.cnn.com/TRANSCRIPTS/0706/15/lkl.01.html |archive-date= June 21, 2007 |url-status= live }}&lt;/ref&gt; [[Sam Neill]], [[Tom Conti]] and [[Charles Grodin]] tested for the part along with Roberts.&lt;ref name=&quot;Pretty Woman&quot; /&gt;&lt;ref&gt;{{Cite web |date=2015-03-23 |title=&quot;Pretty Woman&quot; turns 25: Go behind the scenes with 21-year-old Julia Roberts - CBS News |url=https://www.cbsnews.com/news/pretty-woman-turns-25-go-behind-the-scenes-with-21-year-old-julia-roberts/ |access-date=2023-08-04 |website=www.cbsnews.com |language=en-US}}&lt;/ref&gt; Gere initially refused but when he met with Roberts, she persuaded him and he eventually agreed to play Lewis.&lt;ref&gt;{{cite web |last= |date=March 24, 2015 |title='Pretty Woman' Cast Reunites 25 Years Later - TODAY |url=https://www.youtube.com/watch?v=3Ub6BdLFWyg |url-status=live |archive-url=https://web.archive.org/web/20171116022214/https://www.youtube.com/watch?v=3Ub6BdLFWyg |archive-date=November 16, 2017 |access-date=September 10, 2017 |via=YouTube}}&lt;/ref&gt; He reportedly started off much more active in his role; but Garry Marshall took him aside and said &quot;No, no, no, Richard. In this movie, one of you moves and one of you does not. Guess which one you are?&quot;&lt;ref&gt;{{cite book|last=Tiffin|first=George|title=A Star is Born: The Moment an Actress becomes an Icon|url=https://books.google.com/books?id=OZeQCgAAQBAJ|year=2015|publisher=Head of Zeus|isbn=978-1-78185-936-0|page=[https://books.google.com/books?id=OZeQCgAAQBAJ&amp;pg=PT493 493]}}&lt;/ref&gt;

267 Edgewater was used as a filming location for the 1997 feature film ''[[Cop Land]]'', starring [[Sylvester Stallone]], [[Robert De Niro]], [[Harvey Keitel]] and [[Ray Liotta]].&lt;ref&gt;[[Trip Gabriel|Gabriel, Trip]]. [https://www.nytimes.com/1997/08/10/movies/stallone-seeks-a-serious-turn-for-the-better.html &quot;Stallone Seeks a Serious Turn for the Better&quot;], ''[[The New York Times]]'', August 10, 1997. Page 2 of 6. Accessed December 17, 2011. &quot;He gained the weight with the aid of the Canadian Pancake House on Second Avenue in Manhattan, where he often had breakfast last fall before showing up on the set in Edgewater, N.J. His waist size ballooned to a 39.&quot;&lt;/ref&gt;

268 In 2009, Carpenter starred in production company Red Sparrow's first film, ''Psychosis'', which was released on January 11, 2011. In August 2010, she had a supporting role in [[Sylvester Stallone]]'s ''[[The Expendables (2010 film)|The Expendables]]''.&lt;ref&gt;{{Cite magazine | url = https://variety.com/2009/film/news/austin-carpenter-join-expendables-1118001115/ | title = Austin, Carpenter join 'Expendables' | last = Dodd | first = Stacy | magazine = [[Variety (magazine)|Variety]] | date = March 11, 2009 | access-date = July 10, 2019 | archive-url = https://web.archive.org/web/20190711031431/https://variety.com/2009/film/news/austin-carpenter-join-expendables-1118001115/ | archive-date = July 11, 2019 | url-status = live}}&lt;/ref&gt; After the success of ''The Expendables'', Carpenter was cast in ''Human Factor''&lt;ref&gt;{{Cite web | url = http://www.thehollywoodnews.com/2010/08/22/charisma-carpenter-and-sean-patrick-flanery-join-michael-madsens-the-human-factor/ | title = Charisma Carpenter and Sean Patrick Flanery join Michael Madsen's The Human Factor | last = Heath | first = Paul | website = The Hollywood News | date = August 22, 2010 | access-date = July 10, 2019 | archive-url = https://web.archive.org/web/20180909114204/http://www.thehollywoodnews.com/2010/08/22/charisma-carpenter-and-sean-patrick-flanery-join-michael-madsens-the-human-factor/ | archive-date = September 9, 2018 | url-status = live}}&lt;/ref&gt; and in the indie thriller ''Crash Site''.&lt;ref&gt;{{Cite web | url = http://www.cinemablend.com/new/Charisma-Carpenter-And-Ty-Olsson-Sign-On-For-Crash-Site-20433.html | title = Charisma Carpenter and Ty Olsson Sign on for Crash Site | last = Rich | first = Katey | website = CinemaBlend | date = August 31, 2010 | access-date = October 20, 2010 | archive-url = https://web.archive.org/web/20100904075658/http://www.cinemablend.com/new/Charisma-Carpenter-And-Ty-Olsson-Sign-On-For-Crash-Site-20433.html | archive-date = September 4, 2010 | url-status = live}}&lt;/ref&gt; She appeared in an episode of the seventh season of ''[[Supernatural (U.S. TV series)|Supernatural]]'' alongside ''Buffy'' co-star James Marsters. In 2012, Carpenter landed a recurring role as Rebecca Sewell in the first season of ABC Family drama series ''[[The Lying Game]]''. She was upgraded to a series regular for the second season. Later in the same year, she reprised her role as Lacy in ''[[The Expendables 2]]''.

269 The script was originally pitched to [[Walt Disney Studios (division)|Disney]] subsidiary [[Hollywood Pictures]], with [[David Hoberman]] attached as producer, but Disney placed the film into turnaround,&lt;ref&gt;{{Cite web |last=Archerd |first=Army |date=1996-11-26 |title='George' crew creates urban 'Jungle' |url=https://variety.com/1996/voices/columns/george-crew-creates-urban-jungle-1117863056/ |access-date=2024-09-04 |website=Variety |language=en-US}}&lt;/ref&gt; and sold the project to [[Regency Enterprises|New Regency Productions]], who at that time had a distribution deal with [[Warner Bros.]], with [[Sylvester Stallone]] in mind to star.&lt;ref&gt;{{Cite web |last=Busch |first=Anita M. |date=1996-12-13 |title=Stallone negotiates for 'Negotiator' role |url=https://variety.com/1996/scene/vpage/stallone-negotiates-for-negotiator-role-1117466156/ |access-date=2024-09-04 |website=Variety |language=en-US}}&lt;/ref&gt; Stallone turned the project down and offered the role to [[Samuel L. Jackson]], who had accepted the role.&lt;ref&gt;{{Cite web |last=Busch |first=Anita M. |date=1997-04-03 |title=Jackson 'Negotiates' $5 mil |url=https://variety.com/1997/scene/vpage/jackson-negotiates-5-mil-1117434916/ |access-date=2024-09-04 |website=Variety |language=en-US}}&lt;/ref&gt;

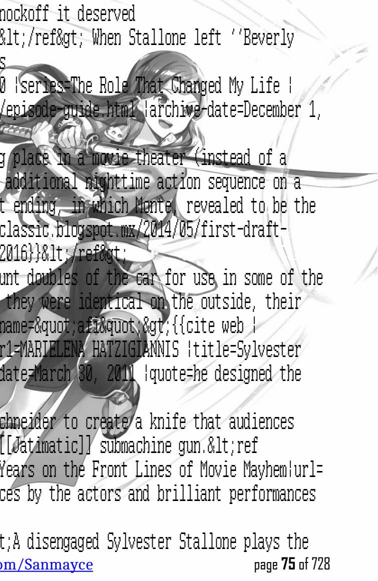
270 Since 2003, the reclusive Johnson has kept well away from the public eye and has concentrated primarily on soundtrack work, scoring numerous documentaries, films, and art installations. {{Not a typo|The The's}} music has featured in a diverse range of cinema over many years, from cult classics such as Jürgen Muschalek's ([[Muscha]]) ''Decoder'' (&quot;Three Orange Kisses from Kazan&quot;) and [[Gregg Araki]]'s ''[[Nowhere (1997 film)|Nowhere]]'' (&quot;Love Is Stronger Than Death&quot;) to big-budget epics such as [[Sylvester Stallone]]'s ''[[Judge Dredd (film)|Judge Dredd]]'' (&quot;Darkness Falls&quot;), prompting Johnson's move into [[film score]] composition.

271 In addition, celebrities returned in full force to the booth, though this proved to be the major criticism of ESPN's first ''MNF'' season. On the opening weekend, Arnold Schwarzenegger, another celebrity-turned-[[Governor of California|California governor]], was in the booth at [[McAfee Coliseum]] in [[Oakland, California]]; before that, [[Jamie Foxx]] appeared at [[FedExField]] in suburban Washington, D.C. Following them, celebrity appearances included [[National Basketball Association|NBA basketball]] superstar [[Dwyane Wade]], [[Basketball Hall of Fame]] player [[Charles Barkley]], [[NASCAR Cup Series]] driver [[Jeff Gordon]], comedian [[Jimmy Kimmel]] (whose opening words to [[Joe Theismann]] were &quot;how's the leg?&quot;), actor [[Sylvester Stallone]], film director [[Spike Lee]], [[hip hop]] artist [[Jay-Z]], and ''MNF'' theme singer [[Hank Williams Jr.]]

272 Sometimes, a [[composer]] may unite with a [[Film director|director]] by composing the score for many films of a same director. [[John Williams]]' professional relationship with [[Steven Spielberg]] and [[George Lucas]] is one of the most prominent in film history, with Williams scoring all but five of Spielberg's films, and all the installments of both of Lucas' blockbuster franchises (''[[Star Wars]]'' and ''[[Indiana Jones]]''); Williams won all five of his Oscars in his collaborations with the two. Additionally, [[Danny Elfman]] did the score for all the movies directed by [[Tim Burton]], with the exception of ''[[Ed Wood (film)|Ed Wood]]'' (score by [[Howard Shore]]) and ''[[Sweeney Todd: The Demon Barber of Fleet Street]]'' (score by [[Stephen Sondheim]]). Other documented instance of director-composer relationships includes: [[Bernard Herrmann]] with [[Alfred Hitchcock]]; [[Jerry Goldsmith]] with [[Joe Dante]] and [[Franklin Schaffner]]; [[Ennio Morricone]] with [[Sergio Leone]], [[Mauro Bolognini]], and [[Giuseppe Tornatore]]; [[Henry Mancini]] with [[Blake Edwards]]; [[Georges Delerue]] with [[François Truffaut]]; [[Alan Silvestri]] with [[Robert Zemeckis]]; [[Angelo Badalamenti]] with [[David Lynch]]; [[James Newton Howard]] with [[M. Night Shyamalan]]; [[Éric Serra]] with [[Luc Besson]]; [[Patrick Doyle]] with [[Kenneth Branagh]]; [[Dave Grusin]] with [[Sydney Pollack]]; [[Howard Shore]] with [[David Cronenberg]], [[Peter Jackson]], and [[Martin Scorsese]]; [[Carter Burwell]] with [[Joel &amp; Ethan Coen]]; [[Bill Conti]] with [[John G. Avildsen]]; [[Lalo Schifrin]] with [[Don Siegel]], [[Stuart Rosenberg]], and [[Brett Ratner]]; [[Hans Zimmer]] with [[Ridley Scott]] and [[Christopher Nolan]]; [[Harry Gregson-Williams]] with [[Tony Scott]] and [[Andrew Adamson]]; [[Clint Mansell]] with [[Darren Aronofsky]]; [[Dario Marianelli]] with [[Joe Wright]]; [[Trent Reznor]] and [[Atticus Ross]] with [[David Fincher]];&lt;ref&gt;{{cite web|url=http://uproxx.com/movies/2014/10/are-david-fincher-and-trent-reznor-the-next-leone-and-morricone/2/|title=Are David Fincher And Trent Reznor The Next Leone and Morricone?|date=October 4, 2014}}&lt;/ref&gt; [[Steve Jablonsky]] with [[Michael Bay]], [[Mychael Danna]] with [[Ang Lee]] and [[Atom Egoyan]], [[Terence Blanchard]] with [[Spike Lee]]; [[Randy Newman]] with [[John Lasseter]]; [[Thomas Newman]] with [[Sam Mendes]]; [[David Newman (composer)|David Newman]] with [[Danny Devito]], [[Brian Levant]], and [[Stephen Herek]]; [[John Debney]] with [[Jon Favreau]] and [[Garry Marshall]]; [[Gabriel Yared]] with [[Anthony Minghella]]; [[Joe Kraemer (composer)|Joe Kraemer]] with [[Christopher McQuarrie]]; [[Michael Giacchino]] with [[J. J. Abrams]] and [[Brad Bird]]; [[James Horner]] with [[James Cameron]] and [[Ron Howard]]; [[John Barry (composer)|John Barry]] with [[Bryan Forbes]], [[Anthony Harvey]], [[Terence Young (director)|Terence Young]], and [[Guy Hamilton]]; [[Elmer Bernstein]] with [[John Landis]], [[John Sturges]], and [[Robert Mulligan]]; [[Maurice Jarre]] with [[David Lean]], [[Peter Weir]], and [[Georges Franju]]; [[Philip Glass]] with [[Godfrey Reggio]]; [[Cliff Martinez]] and [[David Holmes (musician)|David Holmes]] with [[Steven Soderbergh]]; [[Akira Ifukube]] with [[Ishirō Honda]]; [[A. R. Rahman]] with [[Mani Ratnam]]; [[George Fenton]] with [[Richard Attenborough]], [[Nicholas Hynter]], [[Ken Loach]], and [[Stephen Frears]]; [[Klaus Badelt]] and [[Ernst Reijseger]] with [[Werner Herzog]]; [[Randy Edelman]] with [[Ivan Reitman]] and [[Rob Cohen]]; [[Marc Shaiman]] with [[Rob Reiner]]; [[Elliot Goldenthal]] with [[Julie Taymor]] and [[Neil Jordan]]; [[Rachel Portman]] with [[Beeban Kidron]], [[Lasse Hallström]], and [[Jonathan Demme]]; [[Christophe Beck]] with [[Shawn Levy]]; [[Arthur B. Rubinstein]] and [[David Shire]] with [[John Badham]]; [[John Powell (film composer)|John Powell]] with [[Doug Liman]] and [[Paul Greengrass]]; [[Trevor Rabin]] with [[Renny Harlin]] and [[Jon Turteltaub]]; [[Harald Kloser]] with [[Roland Emmerich]]; [[David Arnold (composer)|David Arnold]] with [[Michael Apted]] and [[John Singleton]]; [[Michael Kamen]] with [[Richard Donner]], [[John McTiernan]], and [[Terry Gilliam]]; [[Jorge Arriagada]] with [[Raúl Ruiz (director)|Raúl Ruiz]]; [[Zbigniew Preisner]] with [[Krzysztof Kieślowski]]; [[Mark Isham]] with [[Alan Rudolph]] and [[Robert Redford]]; [[Basil Poledouris]] with [[John Millius]]; [[Joseph Trapanese]] with [[Joseph Kosinski]]; [[Jonny Greenwood]] and [[Jon Brion]] with [[Paul Thomas Anderson]]; [[Brian Tyler]] with [[Justin Lin]] and [[Sylvester Stallone]]; [[John Ottman]] with [[Bryan Singer]]; [[Marco Beltrami]] with [[Wes Craven]] and [[Guillermo del Toro]]; [[Tyler Bates]] with [[James Gunn]], [[Zack Snyder]], and [[Rob Zombie]]; [[Pino Donaggio]] with [[Brian De Palma]]; and [[Alexandre Desplat]] with [[Wes Anderson]], [[Roman Polanski]], and [[George Clooney]].&lt;!--there, better! I fixed a few changes, thanks--&gt;

273 In 1993, Bana made his television debut on [[Steve Vizard]]'s late-night talk show, ''[[Tonight Live with Steve Vizard|Tonight Live]]''.&lt;ref name=&quot;BIO&quot;/&gt; His performance gained the attention of producers from the [[sketch comedy]] series, ''[[Full Frontal (Australian TV series)|Full Frontal]]'', who invited him to join the show as a writer and performer. During his four years

```
    on the show, Bana wrote much of his own material, and based some of his characters on members of his family. His impressions of [[Columbo (TV series)¦Columbo]], [[Arnold Schwarzenegger]], [[Sylvester Stallone]], [[Tom Cruise]] and Australian TV personality [[Ray Martin (television presenter)¦Ray Martin]] made Bana popular with the show's audience.&lt;ref name=&quot;:26&quot;&gt;Devlyn, Darren. &quot;First Impressions&quot; . ''TV Weekly''. 10 February 1993.&lt;/ref&gt; This success led him to record the comedy album ''[[Out of Bounds (Eric Bana album)¦Out of Bounds]]'' in 1994 and to host his own television special, titled ''Eric'', in 1996. The show, a collection of sketches featuring everyday characters, prompted him to launch a sketch comedy series ''The Eric Bana Show''. The series, written and performed by Bana, featured skits, stand-up and celebrity guests, but failed to attract a substantial audience and was cancelled after only eight episodes due to low ratings.&lt;ref name=&quot;Under&quot;&gt;Halfpenny, Kate. &quot;Under the Gun&quot;. ''Who Magazine''. 8 August 2000.&lt;/ref&gt; Even so, in 1997, he received a [[Logie Award]] for &quot;Most Popular Comedy Personality&quot; for his work on the show.&lt;ref name=&quot;management&quot;&gt;{{cite web¦year=2014¦title=Eric Bana¦url=http://www.laurenbergman.com.au/eric_bana.htm¦url-status=dead¦archive-url=https://web.archive.org/web/20070828193841/http://www.laurenbergman.com.au/eric_bana.htm¦archive-date=28 August 2007¦access-date=10 June 2016¦website=Lauren Bergman Management}}&lt;/ref&gt;
274 In 2023, he bought a Hidden Hills estate from Sylvester Stallone for $17{{spaces}}million.&lt;ref&gt;{{cite web ¦ url=https://robbreport.com/shelter/celebrity-homes/sylvester-stallone-house-hidden-hills-john-fogerty-1235333522/ ¦ title=Sylvester Stallone Sells $17 Million California Estate to John Fogerty ¦ date=October 10, 2023 }}&lt;/ref&gt;
275 Other successful pictures included the [[Fiddler on the Roof (film)¦1971 screen version]] of ''[[Fiddler on the Roof]]''. However, [[Man of La Mancha (film)¦the 1972 film version of ''Man of La Mancha'']] was a failure. New talent was encouraged, including [[Woody Allen]], [[Robert Altman]], [[Sylvester Stallone]], [[Saul Zaentz]], [[Miloš Forman]], and [[Brian De Palma]]. With UA being the distributor for Woody Allen's ''[[Bananas (film)¦Bananas]]'' (1971),&lt;ref&gt;{{Cite web ¦title=AFI{{!}}Catalog ¦url=https://catalog.afi.com/Catalog/moviedetails/54031 ¦access-date=2023-10-09 ¦ website=catalog.afi.com}}&lt;/ref&gt; it started as the distributor for a series of Woody Allen films. With the James Bond, Pink Panther, and Woody Allen films, UA had a series of films based on well known characters in the 1970s.
276 In August 2021, he was confirmed to be starring in the upcoming ''[[Expend4bles]]'' film.&lt;ref&gt;{{Cite web¦title=New 'The Expendables' Movie Set to Star 50 Cent, Megan Fox, Sylvester Stallone, Jason Statham, and More¦url=https://www.complex.com/pop-culture/jason-statham-sylvester-stallone-50-cent-megan-fox-to-star-in-new-expendables-movie¦access-date=September 8, 2021¦website=Complex¦ language=en}}&lt;/ref&gt;
277 * In the 1991 film ''[[Oscar (1991 film)¦Oscar]]'', [[Sylvester Stallone]]'s character asks, &quot;Why don't you phone it in to Walter Winchell?&quot;
278 The club has been featured in several documentary and semi-documentary films such as ''Guadalajara 70,''&lt;ref&gt;Felipe Nepomuceno, ''Guadalajara 70,'' 2002.&lt;/ref&gt; ''Uma história de futebol,''&lt;ref&gt;Paulo Machline, ''Uma história de futebol,'' 1998.&lt;/ref&gt; ''Dogão calabresa,''&lt;ref&gt;Pedro Asbeg, ''Dogão calabresa,'' 2002.&lt;/ref&gt; and ''Boleiros.''&lt;ref&gt;Ugo Giorgetti, ''Boleiros,'' 1998.&lt;/ref&gt; Former players have also been published, most notably Pelé,&lt;ref&gt;Aníbal Massaini Neto, ''[[Pele Eterno¦Pelé Eterno]],'' 2004.&lt;/ref&gt;&lt;ref&gt;Carlos Hugo Christensen, ''O Rei Pelé,'' 1963.&lt;/ref&gt;&lt;ref&gt;Eduardo Escorel and Luiz Carlos Barreto, ''Isto é Pelé,'' 1974.&lt;/ref&gt; but others have appeared in films such as ''Ginga.''&lt;ref&gt;Hank Levine, Marcelo Machado and Tocha Alves, ''Ginga,'' 2004.&lt;/ref&gt; Santos was the featured club in the film ''Asa Branca: Um Sonho Brasileiro,'' a story of a modest but talented soccer player for Santos who reaches stardom.&lt;ref&gt;Djalma Limongi Batista, ''Asa Branca: Um Sonho Brasileiro,'' 1981.&lt;/ref&gt; Pelé appeared, alongside other footballers of the 1960s and 1970s, with [[Michael Caine]], and [[Sylvester Stallone]], in the 1981 film ''[[Escape to Victory]],'' about an attempted escape from a [[World War II]] German [[POW Camp]].&lt;ref&gt;{{cite web ¦url=http://www.escapetovictory.spodrum.co.uk/ ¦title=Escape to Victory is the greatest football film ever made. ¦publisher=The [[Escape to Victory]] Website ¦date=22 August 2011 ¦access-date=22 August 2011 ¦archive-date=16 August 2011 ¦archive-url=https://web.archive.org/web/20110816002243/http://www.escapetovictory.spodrum.co.uk/ ¦url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite web ¦url=http://www.hyperhistory.org/index.php?option=displaypage&amp;Itemid=714&amp;op=page ¦title=The Game of Death: playing soccer with the Nazis ¦publisher=Hyper History ¦date=22 August 2011 ¦access-date=22 August 2011 ¦url-status=dead ¦archive-url=https://web.archive.org/web/20110726163720/http://www.hyperhistory.org/index.php?option=displaypage&amp;Itemid=714&amp;op=page ¦archive-date=26 July 2011 }}&lt;/ref&gt; The club has become a symbol of ''O Jogo Bonito'' (English: [[The Beautiful Game]]) in [[Association football culture¦football culture]]. This was largely thanks to the ''Peixe's'' [[Golden Generation¦golden generation]] of the 1960s, the ''[[Os Santásticos¦Santásticos]],'' considered by some the best club team of all times.&lt;ref name=&quot;CO 2003&quot; /&gt;
279 In 2012, he appeared in a supporting role as [[Cinema of the United States¦Hollywood]] producer Lester Siegel in [[Ben Affleck]]'s drama ''[[Argo (2012 film)¦Argo]]'' with Affleck, [[John Goodman]], and [[Bryan Cranston]]. For his performance, he received his fourth [[Academy Award]] nomination, his second for [[Academy Award for Best Supporting Actor¦Best Supporting Actor]], losing to [[Christoph Waltz]] in ''[[Django Unchained]]''.&lt;ref&gt;{{Cite web¦title=&quot;Argo.&quot; AFI Catalog.¦url=https://catalog.afi.com/Catalog/moviedetails/69015¦access-date=July 3, 2023}}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://www.oscars.org/oscars/ceremonies/2013¦title=The 85th Academy Awards {{!}} 2013¦website=www.oscars.org¦date=October 7, 2014 ¦language=en¦access-date=July 3, 2023}}&lt;/ref&gt; He also received nominations for the [[Golden Globe Award]],&lt;ref name=&quot;globes&quot; /&gt; the [[BAFTA Award]],&lt;ref&gt;{{cite web¦url=http://awards.bafta.org/award/2013/film/supporting-actor¦title=BAFTA {{!}} Film {{!}} Supporting Actor in 2013¦website=bafta.org¦access-date=July 3, 2023}}&lt;/ref&gt; and [[Screen Actors Guild Award]].&lt;ref&gt;{{Cite web ¦date=December 12, 2012 ¦title=19th Annual Screen Actors Guild Awards Nominations Announced ¦url=https://www.comingsoon.net/movies/news/97990-19th-annual-screen-actors-guild-awards-nominations-announced ¦access-date=July 4, 2023 ¦website=ComingSoon.net - Movie Trailers, TV &amp; Streaming News, and More ¦language=en-US}}&lt;/ref&gt;&lt;ref name=&quot;:2&quot;&gt;{{Cite web ¦title=The 19th Annual Screen Actors Guild Awards ¦url=https://www.sagawards.org/awards/nominees-and-recipients/19th-annual-screen-actors-guild-awards ¦access-date=July 4, 2023 ¦website=www.sagawards.org}}&lt;/ref&gt; He did receive the [[Screen Actors Guild Award for Outstanding Performance by a Cast in a Motion Picture]].&lt;ref name=&quot;:2&quot; /&gt; That same year, he appeared in the crime drama ''[[Stand Up Guys]]'', opposite [[Al Pacino]] and [[Christopher Walken]].&lt;ref&gt;{{Cite web ¦last=Ebert ¦first=Roger ¦title=Stand Up Guys movie review &amp; film summary (2013) {{!}} Roger Ebert ¦url=https://www.rogerebert.com/reviews/stand-up-guys-2013 ¦access-date=July 4, 2023 ¦website=www.rogerebert.com ¦language=en}}&lt;/ref&gt; The following year he appeared in the comedy ''[[The Incredible Burt Wonderstone]]'' with [[Steve Carell]], [[Steve Buscemi]], [[Olivia Wilde]], and [[Jim Carrey]] and ''[[Grudge Match]]'' with [[Robert De Niro]], [[Sylvester Stallone]], and [[Kim Basinger]].&lt;ref&gt;{{Cite web ¦agency=The Associated Press ¦date=March 13, 2013 ¦title=Alan Arkin and Steve Carell make magic in 'The Incredible Burt Wonderstone' ¦url=https://www.cleveland.com/movies/2013/03/alan_arkin_and_steve_carell_ma.html ¦access-date=July 4, 2023 ¦website=cleveland ¦language=en}}&lt;/ref&gt; He continued to act in supporting roles in films such as the sports drama ''[[Million Dollar Arm]]'' (2014) with [[Jon Hamm]] and the Christmas comedy ''[[Love the Coopers]]'' (2015).&lt;ref&gt;{{Cite web ¦date=November 13, 2015 ¦title=Love the Coopers - Rotten Tomatoes ¦url=https://www.rottentomatoes.com/m/love_the_coopers ¦access-date=July 4, 2023 ¦website=www.rottentomatoes.com ¦language=en}}&lt;/ref&gt;
280 *July 6 - [[Sylvester Stallone]], actor and director
281 *August 9 - [[Alain Dorval]], French voice actor of [[Sylvester Stallone]], from 1976 to 2024 (d [[2024 in film¦2024]]).
282 ¦ screenplay     = [[Sylvester Stallone]]
283 * Sylvester Stallone
284 '''''Cobra''''' is a 1986 American [[action film]] directed by [[George P. Cosmatos]] and written by [[Sylvester Stallone]], who stars in the titular role. The film, loosely based on the novel ''A Running Duck'' by [[Paula Gosling]] (later published as ''Fair Game'' and [[Fair Game (1995 film)¦filmed under that title in 1995]]), co-stars [[Reni Santoni]], [[Brigitte Nielsen]] and [[Andrew Robinson (actor)¦Andrew Robinson]].&lt;ref&gt;{{cite web¦url=http://www.allmovie.com/movie/cobra-v10123¦work=Allmovie¦title=Cobra¦access-date=July 15, 2013¦author=Brenner, Paul¦publisher=Rovi Corporation}}&lt;/ref&gt; ''Cobra'' follows [[Los Angeles Police Department¦Los Angeles police]] [[Police lieutenant¦Lt.]] Marion &quot;Cobra&quot; Cobretti, who investigates a string of violent crimes and also protects a witness targeted by the perpetrators.
285 * [[Sylvester Stallone]] as Lieutenant Marion &quot;Cobra&quot; Cobretti
286 When Sylvester Stallone was signed to play the lead in ''Beverly Hills Cop'', he decided to rewrite the script almost completely, removing nearly all the comedic aspects and turning it into an action movie that he felt was better suited to him. The studio read his revised script and rejected it.&lt;ref&gt;{{cite web ¦title=Round Two With Stallone: Rocky, Beverly Hills Cop, Rambo 4, Elvis, Poe, Horror, Incredibles 2 &amp;... ¦url=http://legacy.aintitcool.com/node/30865 ¦website=Aint It Cool News}}&lt;/ref&gt; The proposed action scenes would have increased the budget far beyond what they planned.&lt;ref name=&quot;Cronin&quot;&gt;{{cite web ¦date= January 16, 2013 ¦first=Brian ¦last=Cronin ¦title=Movie Legends Revealed: Sly Stallone as Axel Foley? ¦url=https://www.cbr.com/movie-legends-revealed-sly-stallone-as-axel-foley/ ¦website=CBR ¦access-date=January 1, 2021}}&lt;/ref&gt; Stallone later channeled his ideas for it into an original script.&lt;ref name=&quot;Rabin&quot;&gt;{{cite web ¦date=September 12, 2014 ¦last=Rabin ¦first=Nathan ¦author-link=Nathan Rabin ¦title=Cobra gave 1986 the Dirty Harry knockoff it deserved ¦url=https://thedissolve.com/features/forgotbusters/749-cobra-gave-the-1980s-the-dirty-harry-knockoff-it-d/ ¦work=The Dissolve ¦access-date=June 3, 2016}}&lt;/ref&gt; When Stallone left ''Beverly Hills Cop'', [[Eddie Murphy]] was brought in to play the lead role.&lt;ref&gt;{{cite episode ¦title=I Took Over A Role From Someone Else And Now I'm Famous ¦url=http://www.thebiographychannel.co.uk/shows/the-role-that-changed-my-life/series-one/episode-guide.html ¦url-status=dead ¦access-date=December 14, 2010 ¦series=The Role That Changed My Life ¦ season=1 ¦number=4 ¦archive-url=https://web.archive.org/web/20101201164810/http://thebiographychannel.co.uk/shows/the-role-that-changed-my-life/series-one/episode-guide.html ¦archive-date=December 1, 2010}}&lt;/ref&gt;
287 Sylvester Stallone's earlier draft of the script contained many differences from later drafts and the final film. These include the opening shootout taking place in a movie theater (instead of a supermarket), during which many more people are killed; Cobra mentioning how he had a girlfriend who was killed by a psychopath he was trying to catch; an additional nighttime action sequence on a boat where Cobra and Ingrid are hiding and are attacked by the Night Slasher's cultists, with Cobra and Gonzalez managing to kill them all; and a different ending, in which Monte, revealed to be the actual leader of the New World, attempts to kill Ingrid at the last second before being killed by Cobretti.&lt;ref&gt;{{cite web ¦url=http://rawrvintageisclassic.blogspot.mx/2014/05/first-draft-screenplay-of-cobra.html ¦title=First Draft Screenplay of Cobra. ¦website=rawrvintageisclassic.blogspot.com ¦date=May 20, 2014 ¦access-date=September 11, 2016}}&lt;/ref&gt;
288 The custom [[Mercury Eight#Third generation (1949-1951)¦1950 Mercury]] driven by Cobretti was actually owned by Sylvester Stallone. The studio produced stunt doubles of the car for use in some of the action sequences, such as the jump from the second floor of the parking garage. The production built three &quot;Cobra cars&quot; for stunt work. Although they were identical on the outside, their moving parts were designed for specific sequences, involving high-speed swipes with other vehicles, 180-degree turns, jumps, and 360-degree spins.&lt;ref name=&quot;afi&quot;&gt;{{cite web ¦ url=https://catalog.afi.com/Catalog/moviedetails/57290 ¦title=Cobra ¦website=AFI ¦access-date=February 28, 2016 }}&lt;/ref&gt;&lt;ref&gt;{{cite web ¦author1=MARIELENA HATZIGIANNIS ¦title=Sylvester Stallone reportedly reunites with stolen car ¦url=https://www.cbsnews.com/news/sylvester-stallone-reportedly-reunites-with-stolen-car/ ¦website=CBS News ¦date=March 30, 2011 ¦quote=he designed the car to fit his role in the movie, and that the car is worth around $250,000}}&lt;/ref&gt;
289 The knife used by the Night Slasher was made for the film by knife designer Herman Schneider.&lt;ref name=&quot;afi&quot; /&gt; Sylvester Stallone asked Schneider to create a knife that audiences would never forget. Cobretti uses a custom [[Colt Gold Cup National Match 1911]], modified to chamber [[9x19mm Parabellum]]. Later in the film, he uses a [[Jatimatic]] submachine gun.&lt;ref name=&quot;afi&quot; /&gt;&lt;ref&gt;{{cite book ¦last= Hunter¦first=Stephen ¦author-link= Stephen Hunter¦date= 1995¦title= Violent Screen: A Critic's 13 Years on the Front Lines of Movie Mayhem¦url= https://books.google.com/books?id=tiKjMVaU0VwC&amp;pg=PT251¦publisher= Bancroft Press¦quote=...Sylvester Stallone's “Cobra,” which features numb performances by the actors and brilliant performances by a Finnish Jati-matic 9mm submachine gun with a laser sighting system...¦isbn=9780963537645}}&lt;/ref&gt;
290 On [[Rotten Tomatoes]], the film has an approval rating of 17% based on 23 reviews, with an average rating of 3.40/10. The website's consensus reads, &quot;A disengaged Sylvester Stallone plays the
```



titular ''Cobra'' with no bite in this leaden action thriller, queasily fixated on wanton carnage and nothing else.&quot;&lt;ref name=&quot;tomatoes&quot;&gt;{{cite web |url= http://www.rottentomatoes.com/m/1004407-cobra/ |title=Cobra |publisher=[[Fandango Media]] | work =[[Rotten Tomatoes]] |access-date=September 11, 2024}}&lt;/ref&gt; On [[Metacritic]], the film has a weighted average score of 25 out of 100 based on 9 critics, indicating &quot;generally unfavorable reviews&quot;.&lt;ref&gt;{{cite web |url=https://www.metacritic.com/movie/cobra |title=Cobra (1986) reviews |work=[[Metacritic]] |publisher=[[CBS Interactive]] |access-date= April 4, 2019}}&lt;/ref&gt; Audiences polled by [[CinemaScore]] gave the film a grade &quot;B&quot; on scale of A to F.&lt;ref name=&quot;CinemaScore&quot;&gt;{{cite web |url= https://www.cinemascore.com/publicsearch/index/title/ |title=COBRA (1986) B |work= [[CinemaScore]] |url-status= dead |archive-url= https://web.archive.org/web/20181220122629/https://www.cinemascore.com/publicsearch/index/title/ |archive-date= December 20, 2018}}&lt;/ref&gt;

291 ''[[TV Guide]]'' stated that &quot;Stallone's character is an empty hulk...the few attempts to provide us with little insights into his character are downright laughable.&quot;&lt;ref&gt;{{cite web | url=http://movies.tvguide.com/cobra/review/111069 |title=Cobra: Review |publisher=[[TV Guide]] |access-date=December 15, 2010 |archive-date=April 2, 2012 |archive-url=https://web.archive.org/web/20120402234503/http://movies.tvguide.com/cobra/review/111069 |url-status=dead }}&lt;/ref&gt; Nina Darnton of ''[[The New York Times]]'' opined that the film &quot;pretends to be against the wanton violence of a disintegrating society, but it's really the apotheosis of that violence....[it] shows such contempt for the most basic American values&quot;,&lt;ref&gt;{{cite news |date=May 24, 1986 |last=Darnton |first=Nina |title=Film: Sylvester Stallone as Policeman, in 'Cobra' |url=https://www.nytimes.com/1986/05/24/movies/film-sylvester-stallone-as-policeman-in-cobra.html |newspaper =[[The New York Times]] |access-date=January 18, 2021}}&lt;/ref&gt; and Vincent Canby called it &quot;disturbing for the violence it portrays, the ideas it represents and the large number of people who will undoubtedly go to see it and cheer on its dangerous hero.&quot;&lt;ref&gt;{{cite web |last1=Canby |first1=Vincent |title=FILM VIEW; INSIDE 'COBRA' MAY DWELL A PUSSYCAT |url=https://www.nytimes.com/1986/06/01/movies/film-view-inside-cobra-may-dwell-a-pussycat.html |website=The New York Times |date=June 1, 1986}}&lt;/ref&gt; Sheila Benson of the ''[[Los Angeles Times]]'' panned the film, saying &quot;''Cobra''&lt;nowiki/&gt;'s pretentious emptiness, its dumbness, its two-faced morality make it a movie that begs to be laughed off.&quot;&lt;ref&gt;{{cite web |date=May 24, 1986 |author=Sheila Benson |title=MOVIE REVIEW : THE 'COBRA' THAT SAVED L.A. |url=https://www.latimes.com/archives/la-xpm-1986-05-24-ca-7438-story.html | website=[[Los Angeles Times]]}}&lt;/ref&gt;

292 ''Cobra'' was nominated for six [[Razzie Awards]], including Worst Picture, Worst Actor (Sylvester Stallone), Worst Actress (Brigitte Nielsen), Worst Supporting Actor and Worst New Star (both for Brian Thompson) and Worst Screenplay.&lt;ref&gt;{{cite web |date=1987 |url=http://www.razzies.com/forum/1986-razzie-nominees-winners_topic336.html |title=1986 RAZZIE® Nominees &amp; 'Winners.' | archive-url=https://web.archive.org/web/20120831204656/http://www.razzies.com/forum/1986-razzie-nominees-winners_topic336.html |archive-date=August 31, 2012 |website=Razzies.com |publisher=Golden Raspberry Award Foundation and John Wilson |access-date=October 18, 2013}}&lt;/ref&gt;

293 In 2019, a television series was in development with [[Robert Rodriguez]] serving as director/creator.&lt;ref&gt;https://www.konbini.com/en/cinema/sylvester-stallone-wants-a-series-remake-of-cult-film-cobra {{Dead link|date=February 2021}}&lt;/ref&gt;&lt;ref&gt;{{cite web |date=September 16, 2019 |last=Davis |first=Erik |title=Sylvester Stallone Teaming with Robert Rodriguez on 'Cobra' Reboot; Is 'Tango &amp; Cash' Next? |url=https://www.fandango.com/movie-news/sylvester-stallone-teaming-with-robert-rodriguez-on-cobra-reboot-is-tango-and-cash-next-753897 |website=Fandango |access-date=September 16, 2019}}&lt;/ref&gt; {{As of|2024}}, it is not known if this project is still in development.

294 * [[Sylvester Stallone filmography]]

295 {{Sylvester Stallone}}

296 [[Category:Films with screenplays by Sylvester Stallone]]

297 For their performances, Anthony Anderson and Christopher Walken were both nominated for Worst Supporting Actor at the [[24th Golden Raspberry Awards]], but they lost to [[Sylvester Stallone]] for ''[[Spy Kids 3-D: Game Over]]''. The Australian

298 In the middle 1970s, films began to also reflect the disenfranchisement brought by the excesses of the past twenty years. A deeply unsettling look at alienation and city life, [[Martin Scorsese]]'s ''[[Taxi Driver]]'' earned international praise, first at the [[Cannes Film Festival]] and then at the [[Academy Awards]], where it was nominated for Best Leading Actor ([[Robert De Niro]]), Best Supporting Actress ([[Jodie Foster]]), Best Score ([[Bernard Herrmann]]), and Best Picture. ''[[All the President's Men (film)|All the President's Men]]'' dealt with the impeachment of [[Richard Nixon]], while Lumet's ''[[Network (1976 film)|Network]]'' portrayed greed and [[narcissism]] in both American society and television media. The film won Oscars for Best Actor ([[Peter Finch]]), Best Actress ([[Faye Dunaway]]), Best Supporting Actress ([[Beatrice Straight]]), and Best Screenplay ([[Paddy Chayefsky]]). Thanks to a stellar cast, experienced director, and a poignant story, ''Network'' became one of the largest critical successes of 1976. Another film, ''[[Rocky]]'', about a clubhouse boxer (played by [[Sylvester Stallone]]) who is granted a world championship title fight won the Best Picture [[Academy Award]] that year. The film also became a major commercial success and spawned four sequels through the rest of the 1970s and 1980s.

299 Hell's Kitchen's gritty reputation had made its housing prices lower than elsewhere in Manhattan. Given the lower costs in the past and its proximity to [[Broadway theatre]]s, the neighborhood is a haven for aspiring actors.&lt;ref&gt;[https://liquidspace.com/us/ny/new-york/hells-kitchen &quot;Hell's Kitchen at a Glance&quot;] {{Webarchive|url=https://web.archive.org/web/20210730171502/https://liquidspace.com/us/ny/new-york/hells-kitchen |date=July 30, 2021 }}, [[LiquidSpace]] {{Retrieved |access-date=25 July 2021}}&lt;/ref&gt;&lt;ref&gt;[https://www.actoraesthetic.com/blog/nyc-for-actors &quot;Top 7 Best Neighborhoods In NYC to Live For Actors&quot;] {{Webarchive|url=https://web.archive.org/web/20210725050908/https://www.actoraesthetic.com/blog/nyc-for-actors |date=July 25, 2021 }} by Maggie Bera, actoraesthetic.com {{Retrieved |access-date=25 July 2021}}&lt;/ref&gt; Many famous actors and entertainers have resided there, including [[Burt Reynolds]], [[Rip Torn]], [[Bob Hope]], [[Charlton Heston]], [[James Dean]], [[Madonna]], [[Jerry Seinfeld]], [[Larry David]], [[Alicia Keys]], and [[Sylvester Stallone]]. This is due in large part to the [[Actors Studio]] on West 44th at which [[Lee Strasberg]] taught and developed [[method acting]].&lt;ref&gt;Hirsch, Foster (1984) ''A Method to their Madness: The History of the Actors Studio''. New York: W. W. Norton. {{ISBN|0-393-01783-4}}&lt;/ref&gt;

300 * [[Sylvester Stallone]] (born 1946), actor who was born and raised in the area&lt;ref&gt;Gilbey, Ryan. [https://www.theguardian.com/film/2016/feb/24/sylvester-stallone-profile-creed-rocky &quot;Sylvester Stallone: the wacky people's champ who battled his own ego; Leading the race for best supporting actor Oscar for his comeback in Creed, this self-made star's creative fortunes have been inseparably bound to his most famous creation, Rocky Balboa&quot;] {{Webarchive|url=https://web.archive.org/web/20230408195719/https://www.theguardian.com/film/2016/feb/24/sylvester-stallone-profile-creed-rocky |date=April 8, 2023 }}, ''[[The Guardian]]'', February 24, 2016. Accessed March 3, 2017. &quot;He was born in Hell's Kitchen, New York City, to an astrologer mother and a father whom Stallone claimed ate raw sparrows and rabbit entrails.&quot;&lt;/ref&gt;

301 * Butkus Stallone, [[Sylvester Stallone]]'s personal pet [[Bullmastiff]] since a puppy; featured in ''[[Rocky]]'' and ''[[Rocky II]]'' as the boxer's &quot;training partner&quot;.&lt;ref&gt;{{cite web|last1=Dicker|first1=Ron|title=Sylvester Stallone's Tribute To His 'Rocky' Dog Butkus Is Just Beautiful|url=https://www.huffingtonpost.com/entry/see-sylvester-stallones-wistful-tribute-to-rocky-dog-and-pet-butkus_us_58de24b2e4b05eae031ee677|website=HuffPost|access-date=16 January 2018|date=31 March 2017}}&lt;/ref&gt;   Stallone credits Butkus for inspiring the screenplay.&lt;ref&gt;{{cite web|last1=Raphael|first1=Raphael|title=Sylvester Stallone's Dog Butkus Was The Main Inspiration For Rocky|url=http://www.esquire.com/uk/culture/film/news/a14026/sylvester-stallones-dog-butkus-helped-launch-the-actor-to-stardom/|website=Esquire|access-date=16 January 2018|date=31 March 2017}}&lt;/ref&gt;

302 In 1989, Pollard had a minor role in ''[[Sleepaway Camp III: Teenage Wasteland]]'' and a larger role (as the inventor of super weapons and a super car) in ''[[Tango &amp; Cash]]'', which also starred [[Kurt Russell]] and [[Sylvester Stallone]]. Also in 1989, he had a two-episode role as the fifth-dimensional imp-villain [[Mr. Mxyzptlk]] in the ''[[Superboy (TV series)|Superboy]]'' TV series.

303 **[[Sylvester Stallone]], actor

304 &quot;Best of What I Got&quot;, from the band's first album, is featured during the credits to the 1989 movie ''[[Tango &amp; Cash]]'' starring [[Sylvester Stallone]] and [[Kurt Russell]].&lt;ref&gt;{{Citation |title=Tango &amp; Cash (1989) - IMDb |url=http://www.imdb.com/title/tt0098439/soundtrack |access-date=March 16, 2021}}&lt;/ref&gt;

305 Supporting cast and characters include neighbor Irwin &quot;Skippy&quot; Handelman ([[Marc Price]]), who has an eternal crush on Mallory; Nick Moore ([[Scott Valentine (actor)|Scott Valentine]]), Mallory's [[Sylvester Stallone]]-esque artist boyfriend; Lauren Miller ([[Courteney Cox]]); and Alex's feminist, artist girlfriend Ellen Reed ([[Tracy Pollan]], whom Michael J. Fox later married, in 1988). In season 3, episode 17, Elyse gave birth to her fourth child, Andrew (who was played by [[Brian Bonsall]] from season 5 onward). Twins Garrett and Tyler Merriman played baby Andrew.

306 After the Ali-Wepner bout, [[Sylvester Stallone]] wrote the script for ''[[Rocky]]'', which was released in theatres in 1976. Like Wepner, (Rocky) Balboa lasts 15 rounds, but unlike Wepner, he actually &quot;[[The distance (boxing)|goes the distance]]&quot;.&lt;ref name=&quot;usa&quot;&gt;{{cite news |last1=Alexander |first1=Bryan |title=2 4 'Chuck': Five things you should know about the real 'Rocky' Chuck Wepner |url=https://www.usatoday.com/story/life/movies/2017/05/02/who-is-real-rocky-chuck-wepner/101158422/ |access-date=January 29, 2018 |work=USA Today |date=May 2, 2017}}&lt;/ref&gt; For years after ''Rocky'' was released, Stallone denied that Wepner provided inspiration for the movie, though he eventually admitted it.&lt;ref name=&quot;ESPN Wepner Recognized&quot;&gt;{{cite web |url=https://www.espn.com/boxing/story/_/page/IamChuckWepner/chuck-wepner-recognized-rocky-fame |title=Chuck Wepner finally recognized for 'Rocky' fame |date=October 25, 2011 |publisher=Espn.go.com |access-date=November 20, 2012}}&lt;/ref&gt;&lt;ref name=&quot;The Real Rocky2&quot;&gt;{{cite video |people=Feuerzeig, Jeff (Director) |date=October 25, 2011 | title=The Real Rocky |medium=Motion picture |publisher=ESPN Films}}&lt;/ref&gt;

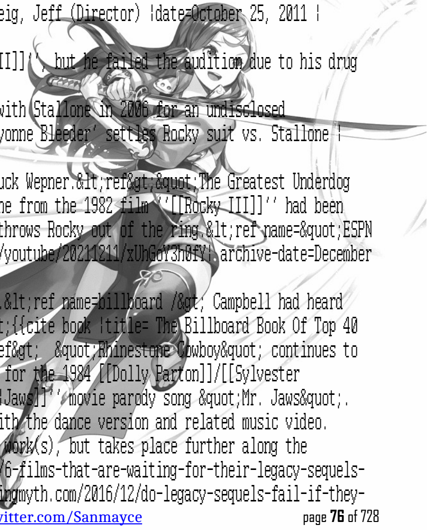
307 After his retirement from boxing, Wepner began abusing drugs. In 1979, Sylvester Stallone wanted to cast Wepner as a sparring partner in ''[[Rocky II]]'', but he failed the audition due to his drug problems.&lt;ref name=&quot;usa&quot; /&gt;

308 In 2003, Wepner sued Sylvester Stallone, seeking payment for his use as the inspiration for ''Rocky'' and the film series. The lawsuit was settled with Stallone in 2006 for an undisclosed amount.&lt;ref name=&quot;The Real Rocky2&quot; /&gt;&lt;ref&gt;{{cite web |url=https://www.espn.com/sports/boxing/news/story?id=2543201 |title='Bayonne Bleeder' settles Rocky suit vs. Stallone | date=August 8, 2006 |website=ESPN.com |access-date=August 11, 2019}}&lt;/ref&gt;

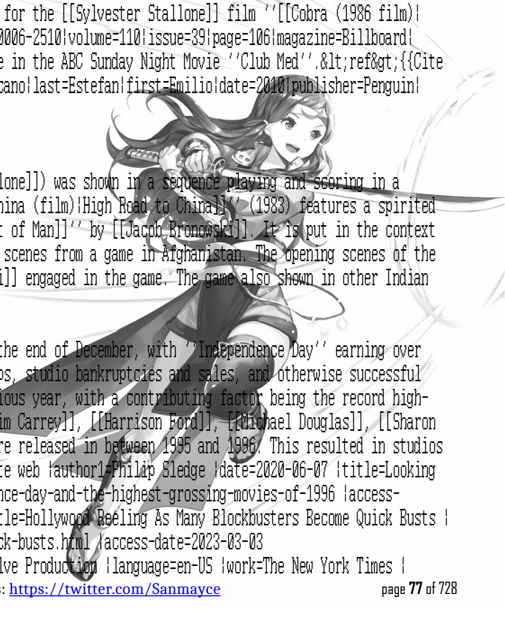
309 * [[Sylvester Stallone]]'s character Rocky Balboa and portions of the [[Rocky (film series)|''Rocky'' film series]] were inspired by the life of Chuck Wepner.&lt;ref&gt;&quot;The Greatest Underdog Story Ever Told | Stallone On Making ROCKY&quot;/https://www.youtube.com/watch?v=PISX0iaMsXc&lt;/ref&gt; For instance, it was speculated that a scene from the 1982 film ''[[Rocky III]]'' had been influenced by Wepner's fight against Andre the Giant, as the movie features a match versus wrestler [[Hulk Hogan]] as &quot;Thunderlips&quot;, who throws Rocky out of the ring.&lt;ref name=&quot;ESPN Wepner Recognized&quot; /&gt;&lt;ref&gt;{{cite web |url=https://www.youtube.com/watch?v=xUhGoY3h0fY |archive-url=https://ghostarchive.org/varchive/youtube/20211211/xUhGoY3h0fY| archive-date=December 11, 2021 |url-status=live|title=Thunderlips.wmv |date=July 2, 2010 |publisher=YouTube |access-date=November 20, 2012}}{{cbignore}}&lt;/ref&gt;

310 &quot;Rhinestone Cowboy&quot; was Campbell's largest-selling single and one of his best-known recordings, initially with over 2 million copies sold.&lt;ref name=billboard /&gt; Campbell had heard songwriter Larry Weiss' version while on tour of Australia in 1974. Both songs were in the October 4, 1975, Hot 100 top 10.&lt;ref name=billboard&gt;{{cite book |title= The Billboard Book Of Top 40 Country Hits: 1944-2006, Second edition|last=Whitburn |first=Joel |author-link=Joel Whitburn |year=2004 |publisher=Record Research |page=66}}&lt;/ref&gt; &quot;Rhinestone Cowboy&quot; continues to be used in TV shows and films, including ''[[Desperate Housewives]]'', ''[[Daddy Day Care]]'', and ''[[High School High]]''. It was the inspiration for the 1984 [[Dolly Parton]]/[[Sylvester Stallone]] movie ''[[Rhinestone (film)|Rhinestone]]''. The main phrase of Campbell's recording was included in [[Dickie Goodman]]'s ''[[Jaws (film)|Jaws]]'' movie parody song &quot;Mr. Jaws&quot;. Campbell also made a techno/pop version of the song in 2002 with UK artists [[Ricardo Autobahn|Rikki &amp; Daz]] and went to the top 10 in the UK with the dance version and related music video.

311 A '''{{visible anchor|legacy sequel|Legacy sequel}}'''&lt;!-- requel or legacy sequel? --&gt; is a work that follows the continuity of the original work(s), but takes place further along the timeline, often focusing on new characters with the original ones still present in the plot.&lt;ref&gt;{{Cite web|url=https://filmschoolrejects.com/6-films-that-are-waiting-for-their-legacy-sequels-5851c0912005/|title=6 Films That Are Waiting for Their Legacy Sequels|date=4 August 2016}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.flickeringmyth.com/2016/12/do-legacy-sequels-fail-if-they-



```
    pander-to-the-fans/|title = Do legacy sequels fail if they pander to the fans?|date = 30 December 2016}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.denofgeek.com/us/movies/creed/251731/creed-2-loses-sylvester-stallone-as-director|title=Creed 2 Loses Sylvester Stallone as Director|date=12 December 2017}}&lt;/ref&gt; Legacy sequels are sometimes also direct sequels that ignore previous installments entirely, effectively [[Retroactive continuity|retconning preceding events]]. ''[[Superman Returns]]'' (2006), ''[[Halloween (2018 film)|Halloween]]'' (2018), ''[[Candyman (2021 film)|Candyman]]'' (2021), ''[[Cobra Kai]]'' (2018-2025), ''[[Blade Runner 2049]]'' (2017), the [[Star Wars sequel trilogy|''Star Wars'' sequel trilogy]], ''[[Ghostbusters: Afterlife]]'' (2021), ''[[Terminator: Dark Fate]]'' (2019), ''[[Tron: Legacy]]'' (2010), ''[[Top Gun: Maverick]]'' (2022), ''[[Doctor Sleep (2019 film)|Doctor Sleep]]'' (2019), ''[[Rocky Balboa (film)|Rocky Balboa]]'' (2006), ''[[Mary Poppins Returns]]'' (2018), ''[[The Matrix Resurrections]]'' (2021), ''[[Indiana Jones and the Kingdom of the Crystal Skull]]'' (2008), and the ''[[Jurassic Park#Jurassic World trilogy|Jurassic World Trilogy]]'' are examples of legacy sequels. Another term for these types of movies is '''{{visible anchor|requel|Requel}}''', meaning reboot sequel, a term originally coined by [[Bruce Campbell]] to describe ''[[Evil Dead 2]]'' (1987) in relation to ''[[The Evil Dead]]'' (1981).&lt;ref&gt;{{Cite web |last=Squires |first=John |date=3 May 2017 |title=Bruce Campbell Finally Settled the 'Evil Dead 2' "Sequel or Remake?" Debate |url=https://bloody-disgusting.com/news/3435456/bruce-campbell-finally-settled-evil-dead-2-sequel-remake-debate |access-date=3 May 2017 |website=[[Bloody Disgusting]] |language=en-US}}&lt;/ref&gt; Film journalist Pamela McClintock describes a requel as something that &quot;exploits goodwill toward the past while launching a new generation of actors and stories&quot;.&lt;ref&gt;{{Cite web |last=McClintock |first=Pamela |date=2016-03-30 |title='Batman v. Superman,' 'Star Wars' and Hollywood's New Obsession With the &quot;Requel&quot; |url=https://www.hollywoodreporter.com/movies/movie-news/batman-v-superman-star-wars-879197/ |access-date=2023-03-19 |website=The Hollywood Reporter |language=en-US}}&lt;/ref&gt; The term was popularized by the film ''[[Scream (2022 film)|Scream]]'' (2022).{{fact|date=February 2024}}
312 ''Get Carter'' eventually garnered a [[cult following]], and further endorsements from directors such as [[Quentin Tarantino]] and [[Guy Ritchie]] led to the film being critically re-evaluated,&lt;ref name=&quot;Crossan-09-12-2009&quot;&gt;{{cite news |last=Crossan |first=Rob|title=Defining Moment: 'Get Carter' rewrites the gangster movie, March 1971 |url=http://www.ft.com/cms/s/0/25adefe8-9c0f-11de-b214-00144feabdc0.html#axzz1pJXFzqcD |archive-url=https://ghostarchive.org/archive/sVxvm |archive-date=10 December 2022 |url-access=subscription |access-date=16 March 2012|newspaper=Financial Times Magazine |date=12 September 2009}}&lt;/ref&gt; with its depiction of class structure and life in 1970s Britain and [[Roy Budd]]'s minimalist [[jazz]] score receiving considerable praise.&lt;ref name=AFI /&gt; In 1999, ''Get Carter'' was ranked 16th on the [[BFI Top 100 British films]] of the 20th century; five years later, a survey of British film critics in ''[[Total Film]]'' magazine chose it as the greatest British film of all time.&lt;ref name=&quot;03-10-2004&quot;&gt;{{cite news |title=''Get Carter'' tops British film poll |url=http://news.bbc.co.uk/1/hi/entertainment/film/3711460.stm |access-date=6 April 2012 |publisher=[[BBC News]] |date=3 October 2004 |archive-url=https://web.archive.org/web/20060515044118/http://news.bbc.co.uk/1/hi/entertainment/film/3711460.stm |archive-date=15 May 2006 |url-status=live |df=dmy-all}}&lt;/ref&gt; A poorly received second remake under the [[Get Carter (2000 film)|same title]] was released in 2000, with [[Sylvester Stallone]] portraying Jack Carter and Caine in a supporting role.
313 Warner Bros., which holds the rights to the pre-1986 MGM library, produced another remake of ''Get Carter'' in 2000 under the [[Get Carter (2000 film)|same title]], starring [[Sylvester Stallone]] as Jack Carter. Originally announced in 1997, [[Tarsem Singh]] and [[Samuel Bayer]] were considered to direct the film before [[Stephen Kay]] signed on, with [[David McKenna (writer)|David McKenna]] writing the script.&lt;ref&gt;{{cite web |last=Karon |first=Paul |title=McKenna gets 'Carter' for WB |url=https://variety.com/1997/film/news/mckenna-gets-carter-for-wb-1116675332/ |work=Variety |date=October 24, 1997 |access-date=March 28, 2022}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Petrikin |first=Chris |title=Musicvid helmer Bayer may 'Get Carter' &amp; Sly |url=https://variety.com/1999/film/news/musicvid-helmer-bayer-may-get-carter-sly-1117502718/ |work=Variety |date=June 3, 1999 |access-date=March 28, 2022}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Petrikin |first=Chris |title=Kay gets 'Carter' to helm Stallone |url=https://variety.com/1999/film/news/kay-gets-carter-to-helm-stallone-1117744322/ |work=Variety |date=July 29, 1999 |access-date=March 28, 2022}}&lt;/ref&gt; As with ''Hit Man'', the film credited Ted Lewis' ''Jack's Return Home'' as its source, not Hodges' film, and again it contains scenes that are directly borrowed from the original, such as the opening train ride.&lt;ref name=&quot;Mitchell-10-7-2000&quot;&gt;{{cite news |last=Mitchell |first=Elvis |title=Movie Review Get Carter (2000) FILM REVIEW; Slimline Stallone, With a Bruising Touch and a Gentle Mutter |url=https://movies.nytimes.com/movie/review?res=980CE5DE153CF934A35753C1A9669C8B63&amp;partner=Rotten%20Tomatoes |access-date=22 February 2012 |newspaper=The New York Times |date=7 October 2000}}&lt;/ref&gt; Michael Caine appears as Cliff Brumby, in what [[Elvis Mitchell]] described as &quot;a role that will increase regard for the original&quot;, speculating that &quot;maybe that was his intention&quot;.&lt;ref name=&quot;Mitchell-10-7-2000&quot;/&gt; [[Mickey Rourke]] plays the villain Cyrus Paice. The remake was compared unfavourably to the original by the majority of reviewers.&lt;ref&gt;{{cite web |title=Get Carter 2000 critics reviews |url=https://www.rottentomatoes.com/m/1100569-get_carter/reviews/?sort=rotten |work=Get Carter |publisher=Flixster, Inc. |access-date=22 February 2012 |archive-url=https://web.archive.org/web/20131229042917/http://www.rottentomatoes.com/m/1100569-get_carter/reviews/?sort=rotten |archive-date=29 December 2013 |url-status=live |df=dmy-all}}&lt;/ref&gt;&lt;ref name=metacritic&gt;{{cite web |title=Get Carter critic reviews |url=https://www.metacritic.com/movie/get-carter/critic-reviews |work=Metacritic |access-date=22 February 2012 |archive-url=https://web.archive.org/web/20140416201338/http://www.metacritic.com/movie/get-carter/critic-reviews |archive-date=16 April 2014 |url-status=live |df=dmy-all}}&lt;/ref&gt; The consensus opinion of critics on Rotten Tomatoes was that it was &quot;a remake that doesn't approach the standard of the original, ''Get Carter'' will likely leave viewers confused and unsatisfied. Also, reviews are mixed concerning Stallone's acting&quot;.&lt;ref&gt;{{cite web|title=Get Carter (2000)|url=https://www.rottentomatoes.com/m/1100569-get_carter|work=Rotten Tomatoes|access-date=20 March 2012|archive-url=https://web.archive.org/web/20120419023646/http://www.rottentomatoes.com/m/1100569-get_carter/|archive-date=19 April 2012|url-status=live|df=dmy-all}}&lt;/ref&gt; It was so badly received on its US release that Warner Bros. decided not to give it a UK theatrical release, anticipating the film would be savaged by British critics and fans.&lt;ref name=&quot;Chibnall-110-2003&quot;&gt;{{cite book |last=Chibnall |first=Steve |title=Get Carter: The British Film Guide 6 |year=2003 |publisher=I.B. Taurus |location=UK |isbn=978-1-86064-910-3 |page=110 |url=https://books.google.com/books?id=b3uW92r5gB0C&amp;q=casting+carter+steve+chibnall&amp;pg=PA26}}&lt;/ref&gt; Elvis Mitchell in ''[[The New York Times]]'' wrote &quot;it's so minimally plotted that not only does it lack subtext or context, but it also may be the world's first movie without even a [[Text (literary theory)|text]]&quot;.&lt;ref name=&quot;Mitchell-10-7-2000&quot;/&gt; Mike Hodges said in 2003 he had still not seen the remake, but was informed by a friend that it was &quot;unspeakable&quot;. His son brought him a DVD of the film back from Hong Kong and he tried to watch it, but the [[DVD region code|region format]] was incompatible &quot;so we put it in the dustbin&quot;.&lt;ref name=&quot;Brooks-2003&quot;&gt;{{cite news |last=Brooks |first=Xan |title=So macho: interview with Mike Hodges |url=https://www.theguardian.com/film/2003/aug/15/edinburghfilmfestival2003.edinburghfilmfestival |newspaper=The Guardian |date=15 August 2003 |access-date=16 February 2012 |archive-url=https://web.archive.org/web/20131002205752/http://www.theguardian.com/film/2003/aug/15/edinburghfilmfestival2003.edinburghfilmfestival |archive-date=2 October 2013 |url-status=live |df=dmy-all}}&lt;/ref&gt; The film was voted the worst remake of all time in 2004 by users of British DVD rental website ScreenSelect (precursor of [[Lovefilm]]).&lt;ref name=&quot;worst remake&quot;&gt;{{cite news |title=Get Carter is 'worst film remake' |url=http://news.bbc.co.uk/1/hi/entertainment/3969245.stm |access-date=22 February 2012 |publisher=BBC News |date=31 October 2004}}&lt;/ref&gt; On 13 February 2001, the remake was released on [[DVD region code#1|Region 1]] DVD by Warner Bros. Home Entertainment.&lt;ref&gt;{{cite web |title=Get Carter (2000) |url=https://www.dvdsreleasedates.com/movies/3102/Get-Carter-(2000).html |website=DVDs ReleaseDates |access-date=20 August 2022}}&lt;/ref&gt;
314 Trying to cater the public of adult action thrillers that were popular in the 1980s, [[Sylvester Stallone]]'s cop-thriller ''[[Cobra (1986 film)|Cobra]]'' (1986) is a thinly-veiled slasher film advertised as an action movie, and sold 13.2 million tickets. The home video market made stars out of [[character actor]]s such as [[Terry O'Quinn]] and [[Bruce Campbell]], whose respective independent horror-thrillers ''[[The Stepfather (1987 film)|The Stepfather]]'' (1987) and ''[[Maniac Cop]]'' (1988) found more support on home video than in theaters. Quinn returned for ''[[Stepfather II]]'' (1989) but chose not to reprise his role in ''[[Stepfather III]]'' (1992), ''[[Destroyer (1988 film)|Destroyer]]'' (1988), while Campbell followed a similar route with a cameo in ''[[Maniac Cop 2]]'' (1990) and no participation in ''[[Maniac Cop III: Badge of Silence]]'' (1993).
315 Reynolds directed ''[[The Last Producer]]'' (2000), starring himself, and was second-billed in [[Renny Harlin]]'s ''[[Driven (2001 film)|Driven]]'' (2001), starring [[Sylvester Stallone]]. He was also in ''[[Tempted (film)|Tempted]]'' (2001), ''[[Hotel (2001 film)|Hotel]]'' (2001) (directed by [[Mike Figgis]]), and ''[[The Hollywood Sign (film)|The Hollywood Sign]]'' (2001).
316 * [[Sylvester Stallone]], actor, director, screenwriter, and producer 1993-1999
317 In 1985, Miami Sound Machine appeared on ''[[The Tonight Show]]'' with Johnny Carson, ''[[Solid Gold (TV series)|Solid Gold]]'', ''[[American Bandstand]]'', ''[[CBS Morning News]]'', Walt Disney specials, and frequent spots on ''MTV''.&lt;ref&gt;{{Cite magazine|first=Vivian|last=Gude|title=Miami Today: International Gateway, Test Ground-and Launch Pad for Latin Music Market in Full Lift-Off|url=https://archive.org/details/bub_gb_9CMEAAAAMBAJ|issn=0006-2510|volume=98|issue=44|page=M-16|magazine=Billboard|date=November 1, 1986|publisher=Nielsen Business Media, Inc.|language=en}}&lt;/ref&gt; The band twice performed on the television presentation of the [[Miss Universe Pageant]] (1984; 1986). The song &quot;Hot Summer Nights&quot; was also released that year and was part of the soundtrack for the film ''[[Top Gun]].'' The song &quot;Suavé&quot; was also recorded as part of the soundtrack for the [[Sylvester Stallone]] film ''[[Cobra (1986 film)|Cobra]]''.&lt;ref&gt;{{Cite magazine|title=The Discography of E.E.|url=https://books.google.com/books?id=GgoEAAAAMBAJ&amp;pg=PA106|issn=0006-2510|volume=110|issue=39|page=106|magazine=Billboard|date=September 26, 1998|publisher=Nielsen Business Media, Inc.|language=en}}&lt;/ref&gt; Miami Sound Machine also made a cameo appearance in the ABC Sunday Night Movie ''Club Med''.&lt;ref&gt;{{Cite book|url=https://books.google.com/books?id=tUrysb-seoQC&amp;pg=PT133|title=Ritmo Al Exito: Como Un Inmigrante Hizo Su Propio Sueno Americano|last=Estefan|first=Emilio|date=2010|publisher=Penguin|isbn=978-1101656068|language=en}}&lt;/ref&gt;
318 * [[Sage Stallone]] (1976-2012), actor, son of [[Sylvester Stallone]]
319 | [[Sylvester Stallone]] || [[Rocky Balboa]] || ''[[Rocky]]''
320 In ''[[Rambo III]]'' (1988), directed by [[Peter MacDonald (film director)|Peter MacDonald]], [[John Rambo]] (played by [[Sylvester Stallone]]) was shown in a sequence playing and scoring in a buzkashi with his [[mujahideen]] friends when suddenly they were attacked by Soviet forces.  The [[Tom Selleck]] film ''[[High Road to China (film)|High Road to China]]'' (1983) features a spirited game of buzkashi. Buzkashi is described at length in Episode 2, &quot;The Harvest of the Seasons&quot;, of the documentary ''[[The Ascent of Man]]'' by [[Jacob Bronowski]]. It is put in the context of the development, by the [[Mongols]], of warfare using the horse and its effect on agricultural settlements. The film includes several scenes from a game in Afghanistan. The opening scenes of the Indian film ''[[Khuda Gawah]]'' (1992), which was filmed in [[Afghanistan]] and [[India]], show actors [[Amitabh Bachchan]] and [[Sridevi]] engaged in the game. The game also shown in other Indian films like ''[[Kabul Express]]'' (2006) and [[Sahasam (2013 film)|''Sahasam'']] (2013).
321 :[[Golden Raspberry Award for Worst Actor|Worst Actor]]: '''[[Sylvester Stallone]]''' - ''[[Rhinestone (film)|Rhinestone]]''
322 The theatrical year of 1996 achieved a $5.8 billion domestic gross milestone and saw a record twelve films earning over $100 million by the end of December, with ''Independence Day'' earning over $300 million. However, the film industry experienced a surplus of releases between 1995 and 1996 that resulted in several box office flops, studio bankruptcies and sales, and otherwise successful films struggling to turn in a profit. The average cost of producing and marketing film rose to $60 million, a 20% increase from the previous year, with a contributing factor being the record high-salaries for stars such as [[Sylvester Stallone]], [[Tom Cruise filmography|Tom Cruise]], [[Arnold Schwarzenegger]], [[Mel Gibson]], [[Jim Carrey]], [[Harrison Ford]], [[Michael Douglas]], [[Sharon Stone]], [[Julia Roberts]], and [[Demi Moore]] that became the subject of industry scrutiny following several high profile flops that were released in between 1995 and 1996. This resulted in studios having to cut their film slate by fifty percent to offset growing production and marketing costs and ensure more profits.&lt;ref&gt;{{Cite web |author1=Philip Sledge |date=2020-06-07 |title=Looking Back On Independence Day And The Highest Grossing Movies Of 1996 |url=https://www.cinemablend.com/news/2547544/looking-back-on-independence-day-and-the-highest-grossing-movies-of-1996 |access-date=2023-03-03 |website=CINEMABLEND |language=en}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Weinraub |first=Bernard |date=1996-07-30 |title=Hollywood Reeling As Many Blockbusters Become Quick Busts |language=en-US |work=The New York Times |url=https://www.nytimes.com/1996/07/30/movies/hollywood-reeling-as-many-blockbusters-become-quick-busts.html |access-date=2023-03-03 |issn=0362-4331}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Weinraub |first=Bernard |date=1996-06-17 |title=Film Glut Prompts Disney to Halve Production |language=en-US |work=The New York Times |
```



url=https://www.nytimes.com/1996/06/17/business/film-glut-prompts-disney-to-halve-production.html |access-date=2023-03-03 |issn=0362-4331}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |title=Study: Film stars are overpaid - UPI Archives |url=https://www.upi.com/Archives/1996/04/08/Study-Film-stars-are-overpaid/2778828936000/ |access-date=2023-03-03 |website=UPI |language=en}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Dutka |first=Elaine |date=1996-12-31 |title=The Money's Where the Action Is |url=https://www.latimes.com/archives/la-xpm-1996-12-31-ca-14282-story.html |access-date=2023-03-03 |website=Los Angeles Times |language=en-US}}&lt;/ref&gt;

323 *[[Sylvester Stallone]] - ''The Square Root''

324 After retiring from the OSS, Juni is thrust back into service when an evil mastermind named Sebastian &quot;The Toymaker&quot; ([[Sylvester Stallone]]) creates a fictional video game called ''Game Over'', which hypnotizes its users. Carmen was sent on a mission to disable the game, but disappeared on Level 4. With the help of his maternal grandfather, Valentin Avellan ([[Ricardo Montalbán|Ricardo Montalban]]), who uses a wheelchair, Juni is sent after Carmen and helps her to disable the game in order to save the world. It is revealed that Sebastian was the one who disabled Valentin in the first place. Instead of avenging his former partner, Valentin forgives Sebastian who is redeemed.

325 | [[Sylvester Stallone]]

326 * [[Sylvester Stallone]]

327 '''''Spy Kids 3-D: Game Over''''' (also known as '''''Spy Kids 3: Game Over''''') is a 2003 American [[spy film|spy]] [[action comedy film]] co-produced, written, shot, edited, composed, and directed by [[Robert Rodriguez]]. It is the sequel to ''[[Spy Kids 2: The Island of Lost Dreams]]'' (2002) and the third installment in the [[Spy Kids (film series)|''Spy Kids'' film series]]. The film stars [[Antonio Banderas]], [[Carla Gugino]], [[Alexa PenaVega|Alexa Vega]], [[Daryl Sabara]], [[Ricardo Montalbán]], [[Holland Taylor]], [[Mike Judge]], [[Cheech Marin]], and [[Sylvester Stallone]].

328 * [[Sylvester Stallone]] as Agent Sebastian &quot;The Toymaker&quot;, a former OSS agent turned supervillain, and the creator of the virtual reality game, Game Over.

329 For his performance as The Toymaker, [[Sylvester Stallone]] earned a [[Golden Raspberry Award for Worst Supporting Actor]] at [[John J. B. Wilson]]'s [[2003 Golden Raspberry Awards]] ceremony.

330 *[[Sylvester Stallone]] - ''[[Police Story (1973 TV series)|Police Story]]''

331 The [[1st Golden Raspberry Awards|first Golden Raspberry Awards]] ceremony was held on March 31, 1981, in John J. B. Wilson's living-room alcove in Hollywood, to honor the perceived worst films of the [[1980 in film|1980 film season]]. To date, [[Sylvester Stallone]] is the most-awarded actor ever, with 12 awards.

332 |[[Sylvester Stallone]]

333 |[[Sylvester Stallone]]

334 The [[Razzie Redeemer Award]] is presented to a former nominee or winner who has subsequently made a comeback from critical or commercial failure. The award was introduced in 2014. Winners include [[Ben Affleck]], [[Sylvester Stallone]], [[Mel Gibson]], &quot;A Safe Hollywood-Haven&quot;, [[Melissa McCarthy]], [[Eddie Murphy]], [[Will Smith]], [[Colin Farrell]], and [[Fran Drescher]].

335 This award is given to movies with particularly high budgets that bombed at the box office. It was awarded in 2015 to ''[[United Passions]],''&lt;ref&gt;{{Cite news|url=http://www.hollywoodreporter.com/news/razzie-awards-mock-fifty-shades-870615|title=Razzie Awards Mock 'Fifty Shades of Grey,' Redeem Sylvester Stallone|newspaper=The Hollywood Reporter|access-date=February 24, 2017|language=en|archive-date=February 25, 2017|archive-url=https://web.archive.org/web/20170225051814/http://www.hollywoodreporter.com/news/razzie-awards-mock-fifty-shades-870615|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{Citation|last=Razzie Channel|title=36th Razzie's - Barry L Bumstead Award Goes To…United Passions!|date=April 20, 2016|url=https://www.youtube.com/watch?v=Nl_u_re1-Y0| archive-url=https://ghostarchive.org/varchive/youtube/20211030/Nl_u_re1-Y0| archive-date=2021-10-30|access-date=February 24, 2017}}{{cbignore}}&lt;/ref&gt; to ''[[Misconduct (film)|Misconduct]]'' in 2016,&lt;ref&gt;{{cite web|last1=Heller|first1=Corrine|title=Razzies 2017 Winners Announced: Batman v Superman &quot;Loses&quot; Worst Picture But Still &quot;Wins&quot; Big|url=http://www.eonline.com/news/831927/razzies-2017-winners-announced-batman-v-superman-loses-worst-picture-but-still-wins-big|website=E Online|date=February 25, 2017|publisher=E!|access-date=February 25, 2017|archive-date=March 1, 2017|archive-url=https://web.archive.org/web/20170301084932/http://www.eonline.com/news/831927/razzies-2017-winners-announced-batman-v-superman-loses-worst-picture-but-still-wins-big|url-status=live}}&lt;/ref&gt; in 2017 to ''[[CHiPs (film)|CHiPs]]'' and in 2018 to ''[[Billionaire Boys Club (2018 film)|Billionaire Boys Club]]''.

336 |[[Sylvester Stallone]]

337 |[[Sylvester Stallone]]

338 Pitt had established himself as an in-demand and highly-paid film star following successes with ''[[Interview with the Vampire (film)|Interview with the Vampire]]'' (1994) and ''Legends of the Fall'', but Fincher had not considered him to portray Mills because he felt Pitt did not fit the image he had for the character. However, after meeting Pitt, Fincher believed he was charismatic and likeable enough that he could keep the audience on Mills's side regardless of the character's actions.&lt;ref name=&quot;EWSep1995Kopelson&quot;/&gt;&lt;ref name=&quot;LATimesBradPittBio&quot;/&gt;&lt;ref name=&quot;Cinephilia&quot;/&gt; To help secure Pitt's involvement, Kopelson shortened the pre-production schedule from twelve weeks to five to accommodate Pitt's schedule.&lt;ref name=&quot;LATimesBradPittBio&quot;/&gt; Pitt rejected several offers from other films because he wanted to escape his typecasting as a romantic lead character in favor of something with a more &quot;documentary feel&quot; with urban settings and a focus on dialogue, akin to thriller films such as ''[[The Conversation]]'' (1974).&lt;ref name=&quot;LATimesBradPittBio&quot;/&gt;&lt;ref name=&quot;EWSep1995Kopelson&quot;/&gt; Pitt described Mills as a well-intentioned &quot;idiot&quot; who &quot;speaks before he really knows what he's talking about&quot;. He cut his hair for the role and lost weight to reduce the muscle he developed for ''Legends of the Fall''.&lt;ref name=&quot;LATimesBradPittBio&quot;/&gt; [[Sylvester Stallone]] and [[Denzel Washington]] rejected the role.{{efn|Attributed to multiple references:&lt;ref name=&quot;CastYahooDenzel&quot;/&gt;&lt;ref name=&quot;CastIWDenzel&quot;/&gt;&lt;ref name=&quot;CastAICNStallone&quot;/&gt;&lt;ref name=&quot;CastUproxxMills&quot;/&gt;}}

339 * [[Sylvester Stallone]]

340 '''''Antz''''' is a 1998 American animated [[adventure comedy film]] produced by [[DreamWorks Animation]] (as its debut film) and [[Pacific Data Images]], and distributed by [[DreamWorks Pictures]]. It was directed by [[Eric Darnell]] and [[Tim Johnson (film director)|Tim Johnson]] (in their feature directorial debuts) from a screenplay written by [[Todd Alcott]] and the writing team of [[Chris Weitz|Chris]] and [[Paul Weitz (filmmaker)|Paul Weitz]]. The film stars the voices of [[Woody Allen]], [[Sharon Stone]], [[Jennifer Lopez]], [[Sylvester Stallone]], [[Christopher Walken]], [[Dan Aykroyd]], [[Anne Bancroft]], [[Danny Glover]] and [[Gene Hackman]]. Some of the main characters share facial similarities with the actors who voice them.&lt;ref&gt;{{cite web |url=http://dvd.ciao.co.uk/Antz_DVD__Review_5030989 |title=Antz DVD - Review - Just a big kid |work=ciao! |date=January 30, 2001 |access-date=July 13, 2010 |archive-url=https://web.archive.org/web/20150714030815/http://dvd.ciao.co.uk/Antz_DVD__Review_5030989 |archive-date=July 14, 2015 |url-status=dead |df=mdy-all }}&lt;/ref&gt; The film involves an anxious worker ant, Z (Allen), who falls in love with Princess Bala (Stone). When the arrogant General Mandible (Hackman) attempts to seize control of the ant colony, Z must combine his desire for purpose with his inner strength to save everyone.

341 * [[Sylvester Stallone]] as Corporal Weaver, a brave soldier ant and Z's best friend who becomes Azteca's boyfriend.

342 During this phase, she made her stage debut in 1987 at the age of 22 in the off-Broadway play ''Daughters'', playing Cetta. The role earned her rave reviews and the [[Theatre World Award]] for outstanding debut on stage. She continued to take roles off-Broadway in the plays ''Beirut'' in 1987, ''Sharon and Billy'' in 1988, and ''[[What the Butler Saw (play)|What the Butler Saw]]'' in 1989. She also portrayed Rosa Delle Rose in ''[[The Rose Tattoo]]'' at the [[Williamstown Theatre Festival]]. She took more prominent roles opposite [[Sylvester Stallone]] in the comedy ''[[Oscar (1991 film)|Oscar]]'' (1991), and [[Nicolas Cage]] in the [[erotic thriller]] ''[[Zandalee]]'' (1991).

343 In 1987, she played the sensible, intelligent 18-year-old daughter of [[Patty Duke]]'s lead character in the short-lived [[Fox Broadcasting Company|Fox]] comedy ''[[Karen's Song]],'' and had a guest-star role in an episode of ''[[Night Court]]''. In 1988, she made a short [[Outrageous Okona|guest appearance]] in the ''[[Star Trek: The Next Generation]]'' episode [[The Outrageous Okona]] as Lt. Robinson. In 1989, she guest-starred in an episode of ''[[Quantum Leap (1989 TV series)|Quantum Leap]],'' &quot;Star Crossed&quot;, as the main character's mathematician/scientist future wife; and guest-starred as a nude beauty pageant winner in an episode of ''[[L.A. Law]]''. That year she also made her motion picture debut with a minor role as a young opportunistic actress in ''[[The Big Picture (1989 film)|The Big Picture]],'' starring [[Kevin Bacon]]. She then played [[Sylvester Stallone]]'s younger sister, a dancer, in the big-budget police action-comedy ''[[Tango and Cash]]'', also starring [[Kurt Russell]]; it was a critical and box office disappointment.

344 Basinger played the role of wife in two 2013 films - the independent drama ''[[Third Person (film)|Third Person]]'', with [[Liam Neeson]] and [[Olivia Wilde]], and the sports comedy ''[[Grudge Match]]'', with [[Robert De Niro]] and [[Sylvester Stallone]]. Critic Odie Henderson, describing Basinger in his review for the latter film, remarked that she &quot;looks stunning at 60 and provides the film's sole voice of reason&quot;.&lt;ref&gt;{{cite web |url=https://www.rogerebert.com/reviews/grudge-match-2013 |title=Grudge Match |first=Odie |last=Henderson |date=25 December 2013 |access-date=6 February 2019 |work=[[RogerEbert.com]]}}&lt;/ref&gt; She subsequently took on the role of mother in the independent drama ''[[4 Minute Mile]]'' (2014) as well as the part of a woman who, after a miscarriage, sets out on a dangerous quest to obtain a child in the likewise independent production ''The 11th Hour'' (also 2014), which was released for VOD.&lt;ref&gt;{{cite news |url=https://www.latimes.com/entertainment/movies/la-et-mn-kim-basinger-11th-hour--20150611-story.html |title=Kim Basinger made the time for low-budget '11th Hour' |first=Susan |last=King |author-link=Susan King (journalist) |date=11 June 2015 |access-date=6 February 2019 |newspaper=[[LA Times]]}}&lt;/ref&gt; ''[[IndieWire]]'' felt that Basinger &quot;does what she can with [''The 11th Hour''] material, but that's not much&quot;.&lt;ref&gt;{{cite web |url=https://www.indiewire.com/2015/06/review-the-11th-hour-starring-kim-basinger-255292/ |title=Review: 'The 11th Hour' Starring Kim Basinger |first=Zach |last=Hollwedel |date=3 June 2015 |access-date=6 February 2019 |work=[[Indie Wire]] |publisher=[[Penske Business Media]]}}&lt;/ref&gt; In 2016, she had a brief role, as a crooked high-ranking official in the [[United States Department of Justice]], in the crime comedy ''[[The Nice Guys]]'', alongside [[Russell Crowe]] and [[Ryan Gosling]].

345 Levy's eventual replacement, [[Joe Wizan]], was receptive to a sequel, and although other executives remained noncommittal, Giler's development executive, Larry Wilson, began looking for a scriptwriter by mid-1983.&lt;ref name=&quot;BloodyDisgustingRetro&quot;/&gt;&lt;ref name=&quot;latimesJuly241986&quot;/&gt; Wilson came across the script for the in-development science fiction film, ''[[The Terminator]]'' (1984), written by [[James Cameron]]. With Cameron's collaborative scriptwriting efforts alongside [[Sylvester Stallone]] on ''[[Rambo: First Blood Part II]]'' (1985), Wilson was convinced to show the script for ''The Terminator'' to Giler, Hill, and Carroll.&lt;ref name=&quot;NYTimes1985July9&quot;/&gt;&lt;ref name=&quot;LATimesJuly1119886&quot;/&gt; In November 1983, Cameron submitted a 42-page [[film treatment]] for ''Alien{{nbsp}}II''-written in three days-based on Giler and Hill's suggestion of &quot;Ripley and soldiers&quot;.&lt;ref name=&quot;BloodyDisgustingRetro&quot;/&gt;&lt;ref name=&quot;latimesJuly241986&quot;/&gt;{{sfn|Shay|1986|p=7}} The studio had a mixed reaction, one executive calling it a constant stream of horror without character development.&lt;ref name=&quot;BloodyDisgustingRetro&quot;/&gt;&lt;ref name=&quot;latimesJuly241986&quot;/&gt; Negotiations to sell the sequel rights to ''[[Rambo (franchise)|Rambo]]''{{'s}} developers [[Mario Kassar]] and [[Andrew G. Vajna]] failed and the project stalled again.&lt;ref name=&quot;latimesJuly241986&quot;/&gt;

346 Reviewers consistently praised Weaver's performance.&lt;ref name=&quot;VarietyReview&quot;/&gt;&lt;ref name=&quot;RevEbert&quot;/&gt; Benson called her the &quot;white-hot core&quot; around whose &quot;defiant intelligence&quot; and &quot;sensual athleticism&quot; ''Aliens'' was built, and Ripley returned not for vengeance but out of compassion.&lt;ref name=&quot;LAtimesReview&quot;/&gt; Ebert credited Weaver's sympathetic performance with holding ''Aliens'' together.&lt;ref name=&quot;RevEbert&quot;/&gt; Kogan compared her to a more attractive [[John Rambo]] (Sylvester Stallone's action character).&lt;ref name=&quot;ChicagoTribuneKogan&quot;/&gt; Scott agreed, saying Weaver made action stars like Stallone and Schwarzenegger look like male [[pin-up model]]s. He described her as the ultimate adventure heroine, balancing action with femininity and maternal instincts.{{sfn|Scott|1986|p=D9}} [[Pauline Kael]] was critical of the film overall as too &quot;mechanical&quot;, but praised Weaver's physical presence and performance, writing that, without her, ''Aliens'' was a subpar [[B picture]].&lt;ref name=&quot;NewYorkerKael&quot;/&gt;

347 [[File:John Rambo.jpg¦thumb¦upright=0.8¦alt=Sylvester Stallone as Rambo, holding a gun¦Ripley was often compared to the era's masculine heroes, including [[Sylvester Stallone]]'s [[John Rambo]].]]

348 During the early 1980s while preparing for the feature film ''[[Rocky III]]'', [[Sylvester Stallone]] explored the possibility of using a real heavyweight boxer in the role of [[Clubber Lang¦James &quot;Clubber&quot; Lang]] by inviting Earnie Shavers to spar with him. Shavers initially refused to hit Stallone with anything other than a soft jab. This frustrated Stallone, who asked Shavers, &quot;C'mon Earnie, show me something real.&quot; Earnie responded by punching him once near the liver, forcing an immediate retirement; Stallone later said: &quot;that nearly killed me. I went straight to the men's room and threw up&quot;.&lt;ref&gt;{{cite book ¦ last = Romano ¦ first = Frederick V. ¦ title = The boxing filmography: American features, 1920-2003¦ publisher = McFarland &amp; Company ¦ date = August 27, 2004 ¦ page = 171¦ isbn = 978-0-7864-1793-3 }}&lt;/ref&gt; However, according to Rhonda Young, the film's [[casting director]], the reason why he was eventually not chosen for the part is that his voice was too high-pitched and not menacing enough ([[Joe Frazier]] was also considered).&lt;ref&gt;Archived at [https://ghostarchive.org/varchive/youtube/20211211/m8uSM2YQoTA Ghostarchive]{{cbignore}} and the [https://web.archive.org/web/20180409124604/https://www.youtube.com/watch?v=m8uSM2YQoTA Wayback Machine]{{cbignore}}: {{Cite web¦url=https://www.youtube.com/watch?v=m8uSM2YQoTA¦title=Mr. T biography (1999)¦website=[[YouTube]] }}{{cbignore}}&lt;/ref&gt;

349 [[File:Sylvester Stallone 2012.jpg¦thumb¦120px¦[[Sylvester Stallone]]]]

350 * [[Sylvester Stallone]] (born 1946), actor, screenwriter, producer, and director

351 Occasionally, filmmakers interact with fans on the site. [[Sylvester Stallone]] answered numerous questions from fans in the site's message boards while publicizing the release of ''[[Rocky Balboa (film)¦Rocky Balboa]]'' as well as ''[[The Expendables (2010 film)¦The Expendables]].''&lt;ref&gt;{{cite web ¦ title = Round One With Sylvester Stallone Q&amp;A!! ¦ publisher = Ain't It Cool News ¦ date = December 1, 2006 ¦ url = http://www.aintitcool.com/node/30861 ¦ access-date = April 23, 2009 ¦ archive-url = https://web.archive.org/web/20090304014831/http://www.aintitcool.com/node/30861 ¦ archive-date = March 4, 2009 ¦ url-status = live }}&lt;/ref&gt; [[Bruce Willis]] also posted on the website briefly to promote his film ''[[Live Free or Die Hard]]''.&lt;ref&gt;{{cite news¦ title=Lights, Camera, Action Hero ¦publisher=Belfast Telegraph ¦date=June 29, 2007 ¦url=http://www.belfasttelegraph.co.uk/entertainment/film-tv/news/lights-camera-action-hero-13454807.html ¦archive-url=https://archive.today/20120720002336/http://www.belfasttelegraph.co.uk/entertainment/film-tv/news/lights-camera-action-hero-13454807.html ¦url-status=dead ¦archive-date=July 20, 2012 ¦access-date=April 23, 2009 }}&lt;/ref&gt;

352 Actor [[Dustin Hoffman]] spoke nasally and was parodied for his [[method acting]]; [[John Gielgud]] and [[Laurence Olivier]] lamented their friends, and even their own death. [[James Bond]] actor [[Roger Moore]] was depicted &quot;with a wooden delivery&quot; - only his eyebrows moved; Moore quipped, &quot;My acting range has always been something between the two extremes of 'raises left eyebrow' and 'raises right eyebrow'.&quot;&lt;ref&gt;{{cite news ¦title=The quintessential Englishman: what we learned from Sir Roger Moore ¦url=https://www.telegraph.co.uk/men/thinking-man/quintessential-englishman-learned-sir-roger-moore/ ¦archive-url=https://ghostarchive.org/archive/20220112/https://www.telegraph.co.uk/men/thinking-man/quintessential-englishman-learned-sir-roger-moore/ ¦archive-date=12 January 2022 ¦url-access=subscription ¦url-status=live ¦access-date=2 October 2019 ¦work=The Telegraph}}{{cbignore}}&lt;/ref&gt; [[Arnold Schwarzenegger]] was muscle-bound but insecure about the size of his genitals; [[Donald Sinden]] was parodied as also trying to become the greatest Shakespearian actor and get a knighthood. [[Clint Eastwood]] was frequently portrayed as an uncompromising tough guy, and [[Sylvester Stallone]] nearly always appeared dressed up as [[John Rambo]].

353 In addition, future [[Emmy Awards¦Emmy]], [[Golden Globe Awards¦Golden Globe]] and [[Academy Awards¦Academy Award]] winning{{\}}nominated actors [[Jean Stapleton]] (credited) and [[Sylvester Stallone]] (uncredited extra) appear in the film.&lt;ref&gt;{{cite book ¦last=Eichhorn ¦first=Dennis P. ¦author-link=Dennis Eichhorn ¦title=Stallone ¦year=1986 ¦location=Seattle ¦publisher=Turman Pub. Co. ¦isbn=978-0-89872-205-5 ¦page=26 ¦url=https://archive.org/details/stallone0000eich/page/26/mode/2up ¦url-access=limited ¦via=[[Internet Archive#Text collection¦Internet Archive Book Reader]]}}&lt;/ref&gt;

354 * [[Daylight (1996 film)¦''Daylight'' (1996 film)]], an American action film starring Sylvester Stallone

355 Hunks of 1985: [[Sylvester Stallone]], [[William &quot;The Refrigerator&quot; Perry]], [[Robert Redford]], [[Jim McMahon]] and [[Pee-wee Herman]]

356 In 2013, D'Onofrio co-starred in the film ''[[Escape Plan (film)¦Escape Plan]]'', filmed in New Orleans, also starring [[Sylvester Stallone]], [[Arnold Schwarzenegger]], and [[50 Cent]]. D'Onofrio portrayed Lester Clark, deputy director of the Prisons Bureau.&lt;ref&gt;{{cite news¦url=https://variety.com/2012/film/news/d-onofrio-50-cent-enter-the-tomb-1118052301/¦newspaper=Variety¦first1=Jeff¦last1=Sneider¦first2=Justin¦last2=Kroll¦title=D'Onofrio, 50 Cent enter 'The Tomb'¦date=April 4, 2012¦access-date=December 15, 2021}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=https://collider.com/the-tomb-filming-sam-neill/166127¦title=Production Begins on 'The Tomb'¦work=[[Collider (website)¦Collider]]¦access-date=April 25, 2018¦archive-date=January 5, 2020¦archive-url=https://web.archive.org/web/20200105201122/https://collider.com/the-tomb-filming-sam-neill/166127/¦url-status=dead}}&lt;/ref&gt;

357 Russell credited his performance in ''[[Tequila Sunrise (film)¦Tequila Sunrise]]'' (1988) with getting Hollywood to regard him differently.{{citation needed¦date=July 2021}} He starred in ''[[Winter People]]'' (1989) and co-starred with [[Sylvester Stallone]] in ''[[Tango &amp; Cash]]'' (1989).

358 '''Tommy Morrison''' (January 2, 1969 - September 1, 2013) was an American [[professional boxer]] and [[mixed martial artist]] who competed from 1988 to 2009. Best known for his left [[hook (boxing)¦hook]] and formidable punching power, Morrison won the [[World Boxing Organization¦WBO]] [[heavyweight]] title in 1993 with a [[George Foreman vs. Tommy Morrison¦unanimous decision victory]] over [[George Foreman]]. He lost the title in [[Tommy Morrison vs. Michael Bentt¦his second defense]] to [[Michael Bentt]] that same year. Morrison's other boxing highlights include [[Ray Mercer vs. Tommy Morrison¦his fight]] with [[Ray Mercer]] in 1991, [[Tommy Morrison vs. Donovan Ruddock¦and with]] [[Donovan Ruddock]] in 1995. Morrison is also known for his acting career, having starred alongside [[Sylvester Stallone]] in the 1990 film ''[[Rocky V]]'' as [[list of Rocky characters#Tommy Gunn¦Tommy Gunn]]. He retired from boxing in 1996 after testing positive for [[HIV]].

359 Morrison started his professional boxing career on November 10, 1988,&lt;ref name=si&gt;{{cite web¦author=Pat Putnam¦url=http://sportsillustrated.cnn.com/vault/article/magazine/MAG1138531/index.htm¦ archive-url=https://web.archive.org/web/20131103235035/http://sportsillustrated.cnn.com/vault/article/magazine/MAG1138531/index.htm¦url-status=dead¦archive-date=November 3, 2013¦title=Tommy Morrison - 06.21.93¦publisher=Sportsillustrated.cnn.com¦date=1993-06-21¦access-date=2013-09-04}}&lt;/ref&gt; with a first-round knockout of William Muhammad in New York City. Three weeks later, he scored another first-round knockout. In 1989, Morrison had 19 wins and 0 losses, 15 by [[knockout]]. That same year, actor [[Sylvester Stallone]] observed one of Morrison's bouts. Stallone arranged a script reading and cast Morrison in the movie ''[[Rocky V]]'' as [[Tommy Gunn (character)¦Tommy &quot;The Machine&quot; Gunn]],&lt;ref&gt;{{cite news¦title= Rocky V' Has Drama Coaches in Its Corner : Film: Acting teachers are traditionally barred from movie sets. But for Stallone's latest boxing epic, a Studio City couple was allowed to show newcomer Tommy Morrison the ropes, scene by scene and blow by blow.¦work=[[Los Angeles Times]]¦date=November 16, 1990¦url=https://www.latimes.com/archives/la-xpm-1990-11-16-ca-4790-story.html¦access-date=2010-10-29¦first=Gary¦last=Klein}}&lt;/ref&gt; a young and talented protege of the retired [[Rocky Balboa]]. Morrison took a six-month break from boxing to work on the movie in 1990.&lt;ref name=si/&gt; From December 8, 1989, until June 8, 1990, Morrison did not compete in a boxing match. That was due to both injuries and his involvement in ''Rocky V''. In 1991, Morrison won four bouts, including notable victories against opponents [[James Tillis]], the first man to take [[Mike Tyson]] the distance, and former WBC heavyweight champion [[Pinklon Thomas]].

360 [[File:Steve Lott-Sylvester Stallone-Tommy Morrison.jpg¦left¦thumb¦Tommy Morrison (on left), [[Steve Lott]] and [[Sylvester Stallone]] on June 6, 1990]]

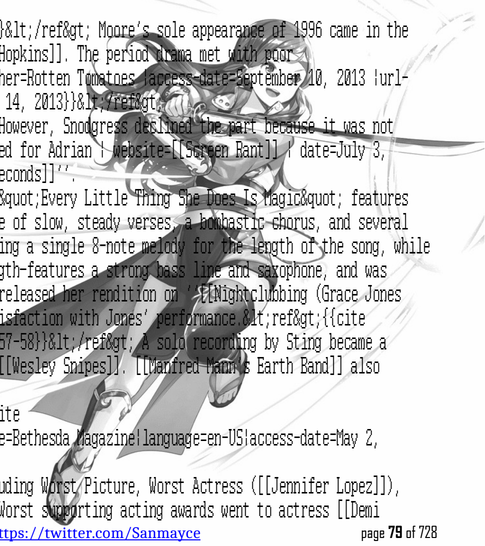
361 Moore's next appearance was a supporting role in the comedy-drama ''[[Roommates (1995 film)¦Roommates]]'' (1995), playing the daughter-in-law of [[Peter Falk]]'s character. Her following film, ''[[Nine Months]]'' (1995), was crucial in establishing her as a Hollywood leading lady.&lt;ref name=&quot;fabulous&quot; /&gt; The romantic comedy, directed by [[Chris Columbus (filmmaker)¦Chris Columbus]] and co-starring [[Hugh Grant]], was poorly reviewed, but a box office success; it remains one of her highest-grossing films.&lt;ref&gt;{{cite web ¦title=Nine Months ¦date=July 12, 1995 ¦ url=https://rottentomatoes.com/m/nine_months/ ¦publisher=Rotten Tomatoes ¦access-date=July 21, 2012 ¦url-status=live ¦archive-url=https://web.archive.org/web/20120720055150/http://www.rottentomatoes.com/m/nine_months/ ¦archive-date=July 20, 2012}}&lt;/ref&gt;&lt;ref&gt;{{cite web ¦title=Nine Months (1995) ¦ url=https://boxofficemojo.com/movies/?id=ninemonths.htm ¦publisher=Box Office Mojo ¦access-date=July 21, 2012 ¦url-status=live ¦archive-url=https://web.archive.org/web/20120725113716/http://boxofficemojo.com/movies/?id=ninemonths.htm ¦archive-date=July 25, 2012}}&lt;/ref&gt;&lt;ref name=&quot;highest gross&quot;&gt;{{cite web ¦title=Julianne Moore Movie Box Office Results ¦url=https://boxofficemojo.com/people/chart/?view=Actor&amp;id=juliannemoore.htm&amp;sort=gross&amp;order=DESC&amp;p=.htm ¦publisher=Box Office Mojo ¦access-date=July 22, 2012 ¦url-status=live ¦archive-url=https://web.archive.org/web/20120416141942/http://www.boxofficemojo.com/people/chart/?view=Actor&amp;id=juliannemoore.htm&amp;sort=gross&amp;order=DESC&amp;p=.htm ¦archive-date=April 16, 2012}}&lt;/ref&gt; Her next release was also a Hollywood production, as Moore appeared alongside [[Sylvester Stallone]] and [[Antonio Banderas]] in the thriller ''[[Assassins (1995 film)¦Assassins]]'' (1995). Despite negativity from critics, the film earned $83.5&amp;nbsp;million worldwide.&lt;ref&gt;{{cite web ¦title=Assassins (1995) ¦date=October 6, 1995 ¦url=https://rottentomatoes.com/m/assassins/ ¦publisher=Rotten Tomatoes ¦access-date=December 2, 2013 ¦url-status=live ¦ archive-url=https://web.archive.org/web/20131216161922/http://www.rottentomatoes.com/m/assassins/ ¦archive-date=December 16, 2013}}&lt;/ref&gt;&lt;ref&gt;{{cite web ¦title=Assassins (1995) ¦ url=http://www.sylvesterstallone.com/film-tv/assassins/ ¦publisher=Sylvester Stallone Online ¦access-date=December 2, 2013 ¦url-status=live ¦archive-url=https://web.archive.org/web/20131202233257/http://www.sylvesterstallone.com/film-tv/assassins/ ¦archive-date=December 2, 2013}}&lt;/ref&gt; Moore's sole appearance of 1996 came in the [[Merchant Ivory Productions¦Merchant Ivory]] film ''[[Surviving Picasso]]'', where she played the artist [[Dora Maar]] opposite [[Anthony Hopkins]]. The period drama met with poor reviews.&lt;ref&gt;{{cite web ¦title=Surviving Picasso ¦date=September 4, 1996 ¦url=https://rottentomatoes.com/m/surviving_picasso/ ¦publisher=Rotten Tomatoes ¦access-date=September 10, 2013 ¦url-status=live ¦archive-url=https://web.archive.org/web/20130814143959/http://www.rottentomatoes.com/m/surviving_picasso/ ¦archive-date=August 14, 2013}}&lt;/ref&gt;

362 According to [[Sylvester Stallone]], Snodgress was the first actress considered for the role of [[Adrian Balboa¦Adrian]] in ''[[Rocky]]''. However, Snodgress declined the part because it was not well-paid.&lt;ref&gt;{{cite web ¦ url=https://screenrant.com/rocky-movie-adrian-actresses-considered/ ¦ title=Rocky: Every Actress Considered for Adrian ¦ website=[[Screen Rant]] ¦ date=July 3, 2021 }}&lt;/ref&gt; ''Rocky'' director [[John G. Avildsen]] cast Snodgress in two of his later films: ''[[A Night in Heaven]]'' and ''[[8 Seconds]]''.

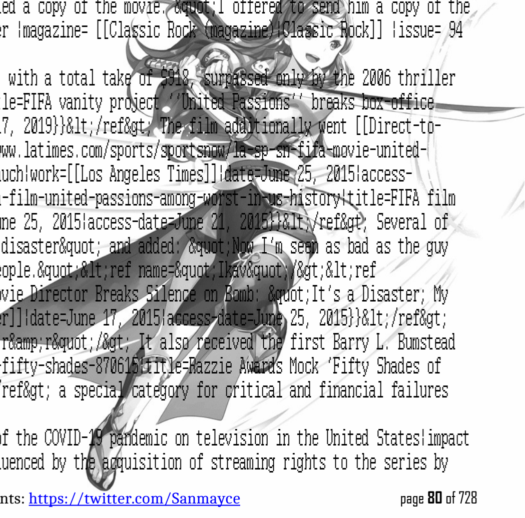
363 The album opens with &quot;Spirits in the Material World&quot;, featuring keyboards dubbed over Summers' [[reggae]]-inspired guitar licks. &quot;Every Little Thing She Does Is Magic&quot; features piano, a strong [[Caribbean music¦Caribbean]] vibe, and an extended non-verbal vocal solo at the end. &quot;Invisible Sun&quot; is a mixture of slow, steady verses, a bombastic chorus, and several guitar solos. &quot;Hungry for You (J'aurais toujours faim de toi)&quot; is sung mostly in French, with the bass and saxophones both repeating a single 8-note melody for the length of the song, while the guitar maintains a steady beat. &quot;[[Demolition Man (song)¦Demolition Man]]&quot;, the band's longest song-almost six minutes in length-features a strong bass line and saxophone, and was written by Sting while staying at [[Peter O'Toole]]'s Irish mansion. The song was originally given to Jamaican singer [[Grace Jones]], who released her rendition on ''[[Nightclubbing (Grace Jones album)¦Nightclubbing]]'' earlier in 1981; the Police then recorded a [[hard rock]] version for ''Ghost in the Machine'' due to their dissatisfaction with Jones' performance.&lt;ref&gt;{{cite magazine¦title=Demolition Men In The Machine: The Police¦first=Chris¦last=Salewicz¦magazine=Creem¦publication-date=April 1982¦pages=26-31, 57-58}}&lt;/ref&gt; A solo recording by Sting became a belated hit in 1993 as the theme song for [[Demolition Man (film)¦the action film of the same title]], starring [[Sylvester Stallone]] and [[Wesley Snipes]]. [[Manfred Mann's Earth Band]] also recorded a version-rearranged and with extensive use of synthesizers-in 1982 for their ''[[Somewhere in Afrika]]'' album.

364 *[[Sylvester Stallone]], American actor, screenwriter, and film director (attended briefly before moving to [[Philadelphia]])&lt;ref&gt;{{Cite web¦url=https://bethesdamagazine.com/bethesda-magazine/sylvester-stallone/¦title=15 Celebrities Who Grew Up Here¦date=April 20, 2015¦website=Bethesda Magazine¦language=en-US¦access-date=May 2, 2019}}&lt;/ref&gt;&lt;ref name=&quot;mocoshow&quot;/&gt;

365 ¦ 28 ¦¦ ''[[Gigli (film)¦Gigli]]'' dominates the [[2003 Golden Raspberry Awards¦Golden Raspberry Awards]], walking away with 6 awards, including Worst Picture, Worst Actress ([[Jennifer Lopez]]), Worst Actor ([[Ben Affleck]]), Worst Director ([[Martin Brest]]), Worst Screenplay (Brest) and worst on-screen couple (Lopez and Affleck). Worst supporting acting awards went to actress [[Demi



```
Moore]] for ''[[Charlie's Angels: Full Throttle]]'' and actor [[Sylvester Stallone]] for ''[[Spy Kids 3-D: Game Over]]''.
366 &lt;/ref&gt; and sold over 800,000 copies, topping the charts in Hong Kong, Mainland China, Taiwan, Singapore and Malaysia.&lt;ref name=cantonese-record&gt;[http://news.bbc.co.uk/1/hi/entertainment/1994254.stm Cantonese record for Wong], BBC, 17 May 2002&lt;/ref&gt; This was her first album after parting from [[Dou Wei]], and her first without any musical collaborations with him since their relationship began. The title track of the album was featured in [[Sylvester Stallone]]'s remake of ''[[Get Carter (2000 film)|Get Carter]]''. The entire album was penned by [[Albert Leung]], who, like Wong, was suffering from a broken heart in his love life at the time. Buddhist philosophy has informed his lyrics since the album, originally as a way to console both Wong and himself. In 2022, Leung picked two of the songs, &quot;Last Blossom&quot; (開到荼靡) and &quot;One Hundred Years of Solitude&quot; (百年孤寂), as his all-time favorites over his prolific career.&lt;ref&gt;{{Cite web |last=自由時報電子報 |date=2022-06-30 |title= (專訪) 林夕公開劈腿王菲陳奕迅 不怕遭非議 - 自由電子報影音頻道 |url=https://video.ltn.com.tw/article/piqgwSuG3fg/PLCvAZ9B-QRHM3BU9tFfccVE4VWRqmTjXl |access-date=2024-06-07 |website=video.ltn.com.tw |language=zh-TW}}&lt;/ref&gt; Wong also became a spokesperson for [[SoftBank Group|JPhone]] in October, 1999, performing in several commercials which aired in Japan. In addition, she began filming for ''[[2046 (film)|2046]]'' in August, a project she would pursue on and off over the next few years when her schedule permitted.
367 Beefcake poses by male actors were used far less frequently than [[Pin-up model|cheesecake]] (pin-up) layouts of actresses. Nevertheless, as early as the 1920s, photographs were taken of such stars as [[Rudolph Valentino]] and [[Ramón Novarro|Ramon Novarro]] to highlight their physical appeal. Male [[Physique photography|physique shots]] of famous stars were even less frequent during the early talking picture era, outside of stars of jungle films such as [[Johnny Weissmuller]] (Tarzan) and [[Buster Crabbe]].  The 1940s saw a rise in shirtless shots of such stars as [[Tyrone Power]], [[Guy Madison]], [[Sterling Hayden]] and [[Victor Mature]]; and in the 1950s movie magazines began running swimsuit shots of actors such as [[Tony Curtis]], [[Rock Hudson]], [[Tab Hunter]], [[Jeff Chandler (actor)|Jeff Chandler]], [[Robert Conrad]] and [[Robert Wagner]] almost as frequently as they did with actresses. This period also included the rise of [[bodybuilding]] magazines, which continue to be popular to the present day, as well as [[Muscleman|musclemen]] movie stars such as [[Steve Reeves]] who were often barely dressed in their action/adventure films. In the 1980s, heavily muscular actors [[Arnold Schwarzenegger]],&lt;ref name=&quot;toronto&quot;&gt;{{cite web |last=Wilner |first=Norman |date=June 14, 2012 |title=Schwarzenegger/ Stallone: The Rise Of Beefcake Cinema |url=https://nowtoronto.com/schwarzenegger-stallone-the-rise-of-beefcake-cinema |archive-url=https://web.archive.org/web/20220305131025/https://nowtoronto.com/schwarzenegger-stallone-the-rise-of-beefcake-cinema |archive-date=March 5, 2022 |access-date=March 5, 2022 |work=[[Now (newspaper)|Now]] |publisher=Media Central Corporation}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://collider.com/eraser-movie-arnold-schwarzenegger-is-it-good-80s-action/|title=How Arnold Schwarzenegger's 'Eraser' Marked the Death of 80s Action|work=[[Collider (website)|Collider]]|publisher=Valnet, Inc.|first=Tom|last=Reimann|date=June 25, 2021|access-date=March 5, 2022}}&lt;/ref&gt; [[Lou Ferrigno]],&lt;ref name=&quot;bostglobe&quot;&gt;{{cite web|url=https://ghostarchive.org/archive/sohPM|title=Review: 'Hercules' labors in vain|work=[[The Boston Globe]]|publisher=Boston Globe Media Partners, LLC|first=Jay|last=Carr|date=August 27, 1983|access-date=March 5, 2022}}&lt;/ref&gt; and [[Sylvester Stallone]]&lt;ref name=&quot;toronto&quot;/&gt; continued to star in beefcake-type action/adventure movies (such as ''[[Conan the Barbarian (1982 film)|Conan the Barbarian]]''&lt;ref name=&quot;toronto&quot;/&gt; or ''[[Hercules (1983 film)|Hercules]]''&lt;ref name=&quot;bostglobe&quot;/&gt;).
368 * [[Sylvester Stallone]]
369 Challenger Films producers Jerry Glover and James Edmunds hire acclaimed English editor Alan Smithee to direct ''Trio'', a blockbuster action film starring [[Sylvester Stallone]], [[Whoopi Goldberg]], and [[Jackie Chan]]. Though only hired due to his inexperience, which the producers believe will make him controllable, Smithee becomes invested in the project.
370 * [[Sylvester Stallone]]
371 | [[Sylvester Stallone]]
372 Tobacco companies have made direct payment to stars for using their [[cigarette]]s in films. Sylvester Stallone received US$500,000 to use [[Brown and Williamson]] tobacco products in five feature films.&lt;ref&gt;[http://legacy.library.ucsf.edu/tid/tmq36b00U.S. Exhibit 21,044] Legacy Tobacco Documents Library&lt;/ref&gt;&lt;ref&gt;[http://legacy.library.ucsf.edu/tid/hlm56b00 Re: agreements between Stallone and Associated Film Promotions] {{Webarchive|url=https://web.archive.org/web/20110614120311/http://legacy.library.ucsf.edu/tid/hlm56b00 |date=June 14, 2011 }} Legacy Tobacco Documents Library&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://legacy.library.ucsf.edu/about/about_collections.jsp#ucbw|title=Master Settlement Agreement Collections|access-date=May 30, 2010|archive-url=https://web.archive.org/web/20100622090229/http://legacy.library.ucsf.edu/about/about_collections.jsp#ucbw|archive-date=June 22, 2010|url-status=live|df=mdy-all}}&lt;/ref&gt;
373 On April 2, 2005, Hogan was inducted into the [[WWE Hall of Fame]] [[WWE Hall of Fame (2005)|class of 2005]] by actor and friend [[Sylvester Stallone]].&lt;ref&gt;{{cite news |url=http://findarticles.com/p/articles/mi_qn4188/is_20050322/ai_n13453252 |archive-url=https://web.archive.org/web/20071111062627/http://findarticles.com/p/articles/mi_qn4188/is_20050322/ai_n13453252 |archive-date=November 11, 2007 |title=Stallone set to induct Hulk Hogan into hall |publisher=Deseret News (Salt Lake City) |access-date=October 25, 2007 |date=March 22, 2005}}&lt;/ref&gt; At [[WrestleMania 21]] on April 3, Hogan came out to rescue [[Nick Dinsmore|Eugene]], who was being attacked by [[Muhammad Hassan (wrestler)|Muhammad Hassan]] and [[Shawn Daivari|Khosrow Daivari]]. The build-up to Hogan's Hall of Fame induction and preparation for his WrestleMania angle was shown on the first season of ''[[Hogan Knows Best]]''. The next night on ''[[WWE Raw|Raw]]'', Hassan and Daivari came out to confront and assault fan favorite [[Shawn Michaels]]. The following week on ''Raw'', Michaels approached Raw General Manager Eric Bischoff demanding a [[Professional wrestling match types#Handicap match|handicap match]] with Hassan and Daivari. Bischoff refused, but told Michaels if he found a partner he would be granted a tag team match. Michaels then made a plea for Hogan to team with him. On the April 18 episode of ''Raw'', Hassan again led an attack on Michaels until Hogan appeared, saving Michaels and accepting his offer. At [[Backlash (2005)|Backlash]], Hassan and Daivari lost to Hogan and Michaels.&lt;ref&gt;{{cite news |first=Scott |last=Zerr |title=Hogan thrilled to be retro |url=http://slam.canoe.ca/Slam/Wrestling/2005/05/01/1021236.html |publisher=SLAM! Sports |date=May 1, 2005 |access-date=October 22, 2007 |url-status=usurped |archive-url=https://archive.today/20120731085353/http://slam.canoe.ca/Slam/Wrestling/2005/05/01/1021236.html |archive-date=July 31, 2012}}&lt;/ref&gt;
374 [[Al Jean]] and [[Mike Reiss]] named Lou after [[Major League Baseball]] player [[Lou Whitaker]],&lt;ref name=&quot;disgrace&quot;/&gt; and Azaria based Lou's voice on that of actor [[Sylvester Stallone]].&lt;ref name=&quot;barbershop&quot;/&gt; Though he has nearly always been African-American,&lt;ref&gt;{{cite web|url=http://www.duffgardens.net/editorials.php?go=editorials/Blacks|title='The Simpsons' and Blacks|work=Springfield Weekly|access-date=February 9, 2010|url-status=dead|archive-url=https://web.archive.org/web/20110723104102/http://www.duffgardens.net/editorials.php?go=editorials%2FBlacks|archive-date=July 23, 2011}}&lt;/ref&gt; he was mistakenly animated with yellow in &quot;[[There's No Disgrace Like Home]]&quot;&lt;ref name=&quot;disgrace&quot;/&gt; and his other first-season appearances. Lou permanently became an African-American character in &quot;[[Bart vs. Thanksgiving]]&quot;, where he was animated to Eddie's voice of [[Harry Shearer]] and vice versa.&lt;ref name=&quot;Vitti, Jon 2002&quot;/&gt;
375 *''[[Over the Top (1987 film)|Over the Top]]'', a 1987 [[Sylvester Stallone]] drama, depicts an [[arm wrestling]] tournament at the Hilton, incorporating footage from an actual tournament that was held there.&lt;ref&gt;{{ubl
376 *[[Sylvester Stallone]]
377 Evans cast [[Al Pacino]], [[Sylvester Stallone]], and [[Harrison Ford]] to portray Dixie Dwyer before Gere was hired.&lt;ref name=&quot;:0&quot; /&gt; [[Richard Pryor]] was considered for the role of Sandman Williams.&lt;ref&gt;{{cite web|last=Evans|first=Bradford|title=The Lost Roles of Richard Pryor|date=1 September 2011|publisher=[[Splitsider]]|url=http://splitsider.com/2011/09/the-lost-roles-of-richard-pryor/|access-date=18 April 2016}}&lt;/ref&gt; Robert Evans wanted to cast his friend [[Alain Delon]] in a two-scene role as [[Lucky Luciano]] but this did not occur.&lt;ref&gt;{{cite news|title=Hollywood: French actor Delon will play Lucky role|author=Beck, Marilyn|work=Chicago Tribune|date=Sep 3, 1982|page=c5}}&lt;/ref&gt; The  role of Luciano was instead portrayed by [[Joe Dallesandro]], starting the dramatic film career for the former [[Warhol superstars|Warhol Superstar]].
378 * ''[[Rambo: Last Blood]]'': American film of 2019, directed by [[Adrian Grunberg]] and with the performances of [[Sylvester Stallone]], [[Matt Cirulnick]] and [[Paz Vega]], among others. It was shot in different locations on the island, such as: different neighborhoods of Santa Cruz de Tenerife and [[San Cristóbal de La Laguna]], areas of [[Puerto de la Cruz]], [[Santa Úrsula]], [[Arico]] and the roads around the Teide National Park, among others places.&lt;ref&gt;[https://www.europapress.es/islas-canarias/noticia-rambo-dejara-tenerife-mas-ocho-millones-euros-20181113140555.html 'Rambo V' dejará en Tenerife más de ocho millones de euros]&lt;/ref&gt;
379 In the film ''[[F.I.S.T. (film)|F.I.S.T.]]'' (1978), [[Sylvester Stallone]] plays Johnny Kovak, a character based on Hoffa.&lt;ref&gt;[https://www.nytimes.com/1978/04/26/archives/screen-fist-drama-of-unionismstallone-returns.html Screen: 'F.I.S.T.', Drama of Unionism:Stallone Returns] {{Webarchive|url=https://web.archive.org/web/20210308172431/https://www.nytimes.com/1978/04/26/archives/screen-fist-drama-of-unionismstallone-returns.html |date=2021-03-08}}, ''[[The New York Times]]'', April 26, 1978&lt;/ref&gt;
380 To &quot;bite the dust&quot; means to die or to lose in a contest or game.  The beginning lyrics set up a scene similar to a [[mobster]] movie, with &quot;machine guns ready to go&quot;. Later lyrics refer to a failed relationship as &quot;another one bites the dust&quot;. The singer is not going to let it get him down, &quot;standing on my own two feet&quot;. The song was used in a preliminary cut of ''[[Rocky III]]'', before being replaced by [[Survivor (band)|Survivor]]'s &quot;[[Eye of the Tiger]]&quot;. &quot;When one of my idols, [[Brian May]], attended one of our shows in Los Angeles in 1984, he brought up that subject&quot;, recalled Survivor guitarist [[Frankie Sullivan]], to whom [[Sylvester Stallone]] had supplied a copy of the movie. &quot;I offered to send him a copy of the tape, which I still own.&quot;&lt;ref&gt;{{cite magazine|first= Dave |last= Ling |title= The stories behind the songs: Eye of the Tiger |magazine= [[Classic Rock (magazine)|Classic Rock]] |issue= 94 |date= July 2006 |page= 36}}&lt;/ref&gt;&lt;ref name=crh/&gt;
381 ''United Passions'' was also a [[box office bomb|box-office bomb]], becoming the second-lowest-grossing film ever in American history, with a total take of $918, surpassed only by the 2006 thriller ''[[Zyzzyx Road]]''.&lt;ref&gt;{{cite web|url=https://avclub.com/article/fifa-vanity-project-united-passions-breaks-box-off-221084|title=FIFA vanity project ''United Passions'' breaks box-office record (not the good kind)|last=Rife|first=Katie|date=June 19, 2015|work=[[The A.V. Club]]|publisher=[[The Onion]]|access-date=April 17, 2019}}&lt;/ref&gt; The film additionally went [[Direct-to-video|directly to DVD]] in France and was not distributed in a number of other European countries.&lt;ref&gt;{{cite news|url=https://www.latimes.com/sports/sportsnow/la-sp-sn-fifa-movie-united-passions-bombs-20150608-story.html|title='United Passions,' which cost FIFA $29 million, has $607 box-office take|author=Austin Knoblauch|work=[[Los Angeles Times]]|date=June 25, 2015|access-date=June 8, 2015}}&lt;/ref&gt;&lt;ref name=&quot;Ikav&quot;&gt;{{cite web|url=http://www.espnfc.com/fifa-world-cup/story/2496658/fifa-film-united-passions-among-worst-in-us-history|title=FIFA film 'United Passions' one of worst in U.S. box office history|author=&lt;!--Not stated--&gt;|work=[[ESPN FC]]|publisher=([[ESPN]])|date=June 25, 2015|access-date=June 21, 2015}}&lt;/ref&gt; Several of the people involved in ''United Passions'' later expressed regret over the film. Director Auburtin called ''United Passions'' &quot;a disaster&quot; and added: &quot;Now I'm seen as bad as the guy who brought AIDS to Africa or the guy who caused the financial crisis ... apparently I am a propaganda guy making films for corrupt people.&quot;&lt;ref name=&quot;Ikav&quot;/&gt;&lt;ref name=&quot;r&amp;r&quot;&gt;{{cite news|url=https://hollywoodreporter.com/news/fifa-movie-director-breaks-silence-802760|title=FIFA Movie Director Breaks Silence on Bomb: &quot;It's a Disaster; My Name Is All Over [This Mess]&quot;|author1=Scott Roxborough |author2=Rhonda Richford |name-list-style=amp|work=[[The Hollywood Reporter]]|date=June 17, 2015|access-date=June 25, 2015}}&lt;/ref&gt; Roth apologized for taking part in the film and admitted that he took the job in ''United Passions'' for the money.&lt;ref name=&quot;r&amp;r&quot;/&gt; It also received the first Barry L. Bumstead Award during the [[36th Golden Raspberry Awards]],&lt;ref&gt;{{Cite web|url=https://www.hollywoodreporter.com/news/razzie-awards-mock-fifty-shades-870615|title=Razzie Awards Mock 'Fifty Shades of Grey,' Redeem Sylvester Stallone|website=The Hollywood Reporter|date=February 28, 2016 |language=en|access-date=January 6, 2020}}&lt;/ref&gt; a special category for critical and financial failures that were not given an eligible release.&lt;ref name=smh/&gt;
382 In January 2022, ''[[Deadline Hollywood|Deadline]]'' reported that the network relaunch had been shelved, partly due to the [[Impact of the COVID-19 pandemic on television in the United States|impact of the COVID-19 pandemic on television and film production]], in addition to ''Yellowstone''{{'}}s ratings success-which had been influenced by the acquisition of streaming rights to the series by
```



[[NBCUniversal]]'s [[Peacock (streaming service)|Peacock]].&lt;ref name=&quot;deadline-pmn-ice&quot;&gt;{{cite web|url=https://deadline.com/2022/01/paramount-movie-network-put-on-ice-as-viacomcbs-nixes-rebrand-plans-1234917830/|title=Paramount Movie Network Put On Ice As ViacomCBS Nixes Rebrand Plans For 'Yellowstone' Network|first=Peter|last=White|work=[[Deadline Hollywood|Deadline]]|date=January 24, 2022|access-date=January 24, 2022}}&lt;/ref&gt; The network also acquired linear rights to the [[Spectrum Originals]] series ''George &amp; Tammy''.&lt;ref name=&quot;deadline-pmn-ice&quot; /&gt; During the Paramount Global upfronts in May 2022, it was announced that ''Yellowstone'' spin-off ''6666'' had been shifted from Paramount+ to Paramount Network, the fifth season of ''Yellowstone'' would premiere in November 2022, and that the first two episodes of Sheridan and [[Sylvester Stallone]]'s Paramount+ series ''[[Tulsa King]]'' would receive special airings as lead-outs for ''Yellowstone''.&lt;ref&gt;{{Cite web |last=White |first=Peter |date=2022-05-18 |title='Yellowstone' Spinoff '6666' Moves From Paramount+ To Paramount Network |url=https://deadline.com/2022/05/yellowstone-spinoff-6666-moves-to-paramount-network-1235026901/ |access-date=2022-05-19 |website=Deadline }}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Pedersen |first=Erik |date=2022-05-18 |title='Yellowstone' Season 5 Premiere Date: Kevin Costner Series Returns To Paramount Network In The Fall |url=https://deadline.com/2022/05/yellowstone-season-5-premiere-date-kevin-costner-paramount-network-1235027462/ |access-date=2022-05-19 |website=Deadline}}&lt;/ref&gt;

383 The [[Brutalist architecture|brutalist]] [[Trinity Square, Gateshead|Trinity Centre Car Park]], which was designed by [[Owen Luder]], dominated the town centre for many years until its demolition in 2010. A product of attempts to regenerate the area in the 1960s, the car park gained an iconic status due to its appearance in the 1971 film ''[[Get Carter]]'', starring [[Michael Caine]]. An unsuccessful campaign to have the structure [[Listed building (United Kingdom)|listed]] was backed by [[Sylvester Stallone]], who played the main role in the [[Get Carter (2000 film)|2000 remake]] of the film.&lt;ref&gt;{{cite web |url=http://www.culture.gov.uk/images/freedom_of_information/106152docs.pdf |title=FOI 106152 released information |access-date=7 November 2012 |url-status=dead |archive-url=https://web.archive.org/web/20120304104753/http://www.culture.gov.uk/images/freedom_of_information/106152docs.pdf |archive-date=4 March 2012 }}&lt;/ref&gt;&lt;ref&gt;{{cite news |last=Collett |first=Christopher |url=http://www.metro.co.uk/news/132154-good-riddance-to-the-gateshead-multi-storey-car-park |title=Good Riddance to the Gateshead Multi-Storey Car Park |work=Metro.co.uk |date=10 April 2008 |access-date=22 May 2012 |url-status=live |archive-url=https://web.archive.org/web/20120930130602/http://www.metro.co.uk/news/132154-good-riddance-to-the-gateshead-multi-storey-car-park |archive-date=30 September 2012 }}&lt;/ref&gt; The car park was scheduled for demolition in 2009, but this was delayed as a result of a disagreement between [[Tesco]], who re-developed the site, and [[Metropolitan Borough of Gateshead|Gateshead Council]].&lt;ref&gt;{{cite news| url=http://news.bbc.co.uk/1/hi/england/tyne/7812682.stm | work=BBC News | title=Historic car park finally closes | date=5 January 2009 | access-date=2 April 2010}}&lt;/ref&gt; The council had not been given firm assurances that Tesco would build the previously envisioned town centre development which was to include a Tesco mega-store as well as shops, restaurants, cafes, bars, offices and student accommodation.&lt;ref name=autogenerated1&gt;{{cite news |author=Pearson, Adrian |url=http://www.chroniclelive.co.uk/north-east-news/todays-evening-chronicle/2009/12/14/council-blocks-get-carter-car-park-demolition-72703-25386958/ |title=Council blocks Get Carter car park demolition |work=ChronicleLive |date=14 December 2009 |access-date=22 May 2012 |url-status=live |archive-url=https://web.archive.org/web/20100501042404/http://www.chroniclelive.co.uk/north-east-news/todays-evening-chronicle/2009/12/14/council-blocks-get-carter-car-park-demolition-72703-25386958/ |archive-date=1 May 2010 }}&lt;/ref&gt;&lt;ref&gt;[http://www.yourtrinitysquare.co.uk/trinity-square-development.asp] {{webarchive|url=https://web.archive.org/web/20080609011624/http://www.yourtrinitysquare.co.uk/trinity-square-development.asp|date=9 June 2008}}&lt;/ref&gt; The council effectively used the car park as a bargaining tool to ensure that the company adhered to the original proposals and blocked its demolition until they submitted a suitable planning application.&lt;ref name=autogenerated1 /&gt; Demolition finally took place in July-August 2010.

384 Andretti appeared in the [[Sylvester Stallone]] film ''[[Driven (2001 film)|Driven]]'' in a cameo. He served as the grand marshal of the 2004 [[Baja 1000]] [[Off-road racing|off-road race]], as seen in the documentary film ''[[Dust to Glory]]''.&lt;ref&gt;{{cite web|last=Krefting |first=Kevin |title=Mario Andretti, the Racer's Racer |url=http://www.ensenadagazette.com/dustglo3_05.html |publisher=[[SPEED TV]] |access-date=December 8, 2011 |url-status=dead |archive-url=https://web.archive.org/web/20101217072903/http://www.ensenadagazette.com/dustglo3_05.html |archive-date=December 17, 2010 }}&lt;/ref&gt; Andretti was featured in the 2007 documentary ''[[A State of Vine (film)|A State of Vine]]'', where he commented on his winemaking activities. In November 2015, he guest starred on an episode of ''[[Jay Leno's Garage]]'', driving Leno in multiple fast cars and talking about his racing career.&lt;ref&gt;{{cite news|title=Mario Andretti on 'Jay Leno's Garage'; Carlos Munoz returning to Andretti Autosport|url=https://www.mcall.com/sports/auto-racing/mc-mario-andretti-jay-leno-20151117-story.html|work=The Morning Call|date=November 17, 2015|access-date=July 30, 2021}}&lt;/ref&gt;

385 Slater starred in the television series ''[[My Own Worst Enemy (TV series)|My Own Worst Enemy]]'' in 2008 and ''[[The Forgotten (TV series)|The Forgotten]]'' in 2009. In 2011 he co-starred in the action film ''[[The River Murders]]'', with [[Ray Liotta]] and [[Ving Rhames]]. Also in 2011, he starred in the television series, ''[[Breaking In (TV series)|Breaking In]]'', which ran for two seasons. Slater co-starred with Ving Rhames in the film ''[[Soldiers of Fortune (2012 film)|Soldiers of Fortune]]'' (2012), and in the [[Sylvester Stallone]] action thriller ''[[Bullet to the Head]]'' (2013), directed by [[Walter Hill (director)|Walter Hill]]. He co-starred in the 2014 television series, ''[[Mind Games (TV series)|Mind Games]]'', which was cancelled after five episodes were aired.&lt;ref&gt;{{cite web|url=https://www.deadline.com/2014/03/abc-mind-games-cancelled-replaced-by-celebrity-wife-swap-resurrection-repeats/|title='Mind Games' Cancelled: ABC Replaces Series With Repeats|last=Andreeva|first=Nellie| date=March 27, 2014|publisher=Deadline.com|access-date=March 28, 2014}}&lt;/ref&gt; He was part of the ensemble in [[Lars von Trier]]'s controversial film, ''[[Nymphomaniac (film)|Nymphomaniac]]''.&lt;ref&gt;{{cite web|last1=Lewis|first1=Hillary|title='Nymphomaniac' N.Y. Premiere: Charlotte Gainsbourg, Christian Slater Support Controversial Director Lars von Trier|url=https://www.hollywoodreporter.com/news/nymphomaniac-ny-premiere-charlotte-gainsbourg-688550|website=Hollywood Reporter|date=March 13, 2014|access-date=January 10, 2016}}&lt;/ref&gt;

386 *March 24: The fight that inspired the movie [[Rocky]]: With a young [[Sylvester Stallone]] sitting at home and watching, [[Muhammad Ali]] retains his world Heavyweight championship with a fifteenth-round knockout over underdog [[Chuck Wepner]], but not without suffering a ninth round knockdown first, in [[Cleveland]].

387 Alagem was the recipient of the Philanthropic Leadership Award from the board of governors of the [[Cedars-Sinai Medical Center]] in 2016.&lt;ref name=&quot;bhcouriercedarssinaiboardofgovernors&quot;&gt;{{cite news|title=Cedars-Sinai Board Of Governors Gala To Honor Adele &amp; Beny Alagem and Sylvester Stallone|work=The Beverly Hills Courier|date=October 10, 2016}}&lt;/ref&gt;

388 In November 2003, [[Sylvester Stallone]] became the second American superstar to participate in the show that year.

389 Other international stars that have appeared in the show through the years are: [[Ricky Martin]], [[t.A.T.u.]], [[Will Smith]] (three times), [[Sylvester Stallone]], [[Arnold Schwarzenegger]], [[Kevin James]], [[Sylvia Saint]], [[Diego Maradona]], [[Christina Aguilera]], [[David Copperfield (illusionist)|David Copperfield]], [[Drew Barrymore]], [[Cameron Díaz]], [[Lucy Liu]], [[Chris Rock]], [[Chayanne]], [[John Leguizamo]], [[N*SYNC]], The [[Backstreet Boys]], [[Gloria Gaynor]], [[Shakira]] and [[Dwayne Johnson|The Rock]]. Worthy of note, although [[Paris Hilton]] was scheduled to appear on the show, she did not show up, leaving him to wait live in front of millions of viewers.

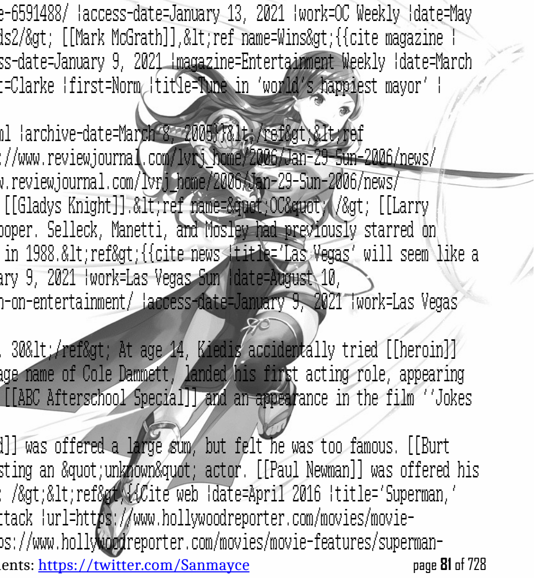
390 During the 1980s and 1990s, he was a co-producer with [[Don Simpson]] of a string of highly successful films for [[Paramount Pictures]]. He first met Simpson at a screening of 1973's ''[[The Harder They Come]]'' at Warner Brothers. The two worked together and created Bruckheimer's first big hit, 1983's ''[[Flashdance]]'', which brought in $95 million.&lt;ref&gt;{{cite news| url=https://movies.nytimes.com/person/83309/Jerry-Bruckheimer/biography | archive-url=https://web.archive.org/web/20071017032130/http://movies.nytimes.com/person/83309/Jerry-Bruckheimer/biography | url-status=dead | archive-date=October 17, 2007 | access-date=May 6, 2010 | department=Movies &amp; TV Dept. | work=[[The New York Times]] | author=Derek Armstrong | date=2007 | title=Movies: Biography forJerry Bruckheimer}}&lt;/ref&gt; He had a number of other hits during that time period, including the ''[[Beverly Hills Cop]]'' films, ''[[Top Gun]]'', and ''[[Days of Thunder]]''. ''Top Gun'' marked his first collaboration with English director [[Tony Scott]], who directed six films for Bruckheimer.&lt;ref name=&quot;RichInterview&quot;&gt;{{cite web|last=Rich|first=Katey|title=Interview: Tony Scott|url=https://www.cinemablend.com/new/Interview-Tony-Scott-13537.html|work=Cinema Blend|access-date=January 3, 2013|date=June 12, 2009}}&lt;/ref&gt; The first ''Beverly Hills Cop'' movie, which was supposed to star [[Sylvester Stallone]], launched [[Eddie Murphy]]'s career and in just five days, became the highest grossing winter release in Paramount's history.&lt;ref&gt;{{cite web|url=https://www.nytimes.com/1984/12/16/movies/exit-stallone-enter-eddie-murphy.html|title=Exit Stallone, enter Eddie Murphy|website=The New York Times|date=16 December 1984|author=Lindsey Gruson}}&lt;/ref&gt; On August 9, 1983, Bruckheimer and Simpson struck a three-year agreement with Paramount to produce theatrical and television projects through his new Simpson/Bruckheimer Productions company.&lt;ref&gt;{{Cite news|date=1983-08-10|title=Par Signs Simpson, Bruckheimer To A 3-Year Exclusivity|page=3|work=[[Variety (magazine)|Variety]]}}&lt;/ref&gt;

391 PSA of 1988: [[Sylvester Stallone]] for Give The Gift Of Literacy

392 Various guest stars have appeared on the show, sometimes portraying themselves.&lt;ref name=Griffin/&gt;&lt;ref&gt;{{cite magazine |last=Tucker |first=Ken |title=Las Vegas |url=https://ew.com/article/2004/04/30/las-vegas/ |access-date=January 9, 2021 |magazine=Entertainment Weekly |date=April 30, 2004}}&lt;/ref&gt;&lt;ref name=OC&gt;{{cite news |last=Lerman |first=Ali |title=Nikki Cox: Humor Makes Her Drown in Good Fortune |url=https://www.ocweekly.com/nikki-cox-humor-makes-her-drown-in-good-fortune-6591488/ |access-date=January 13, 2021 |work=OC Weekly |date=May 9, 2014}}&lt;/ref&gt; Notable guest stars have included [[Alec Baldwin]],&lt;ref name=AP2003/&gt; [[Little Richard]],&lt;ref name=Odds2/&gt; [[Mark McGrath]],&lt;ref name=Wins&gt;{{cite magazine |last=Wheat |first=Alynda |title=NBC wins big with ''Las Vegas'' |url=https://ew.com/article/2004/03/26/nbc-wins-big-las-vegas/ |access-date=January 9, 2021 |magazine=Entertainment Weekly |date=March 26, 2004}}&lt;/ref&gt; [[Sylvester Stallone]],&lt;ref name=Behind/&gt; Las Vegas mayor [[Oscar Goodman]],&lt;ref&gt;{{cite news |last=Clarke |first=Norm |title=Tune in 'world's happiest mayor' |url=http://www.reviewjournal.com/lvrj_home/2005/Jan-14-Fri-2005/news/25663615.html |work=Las Vegas Review-Journal |archive-url=https://web.archive.org/web/20050308103321/http://www.reviewjournal.com/lvrj_home/2005/Jan-14-Fri-2005/news/25663615.html |archive-date=March 8, 2005}}&lt;/ref&gt;&lt;ref name=Closes/&gt; [[Norm Clarke]],&lt;ref&gt;{{cite news |last=Clarke |first=Norm |title=Taking stage fright to new heights |url=http://www.reviewjournal.com/lvrj_home/2006/Jan-29-Sun-2006/news/5594954.html |work=Las Vegas Review-Journal |date=January 29, 2006 |archive-url=https://web.archive.org/web/20060910001319/http://www.reviewjournal.com/lvrj_home/2006/Jan-29-Sun-2006/news/5594954.html |archive-date=September 10, 2006}}&lt;/ref&gt; [[Jewel (singer)|Jewel]],&lt;ref name=Lawrence/&gt; [[Wayne Newton]] and [[Gladys Knight]].&lt;ref name=&quot;OC&quot; /&gt; [[Larry Manetti]] and [[Roger E. Mosley]] made a guest appearance in season 5 as characters named Larry and Roger, who are friends of A.J. Cooper. Selleck, Manetti, and Mosley had previously starred on ''[[Magnum, P.I.]]'', and their appearance together in ''Las Vegas'' marked their first reunion since the ending of ''Magnum, P.I.'' in 1988.&lt;ref&gt;{{cite news |title='Las Vegas' will seem like a 'Magnum, P.I.' reunion |url=https://lasvegassun.com/news/2007/aug/10/las-vegas-will-seem-like-a-magnum-pi-reunion/ |access-date=January 9, 2021 |work=Las Vegas Sun |date=August 10, 2007}}&lt;/ref&gt;&lt;ref&gt;{{cite news |title=Looking In On: Entertainment |url=https://lasvegassun.com/news/2007/sep/11/looking-in-on-entertainment/ |access-date=January 9, 2021 |work=Las Vegas Sun |date=September 11, 2007}}&lt;/ref&gt;

393 Kiedis's father sold drugs, and the two often used [[marijuana]] and [[cocaine]] together.&lt;ref name=Kiedis30&gt;Kiedis, Sloman, p. 30&lt;/ref&gt; At age 14, Kiedis accidentally tried [[heroin]] for the first time, mistaking it for cocaine.&lt;ref name=VH1behindthemusic/&gt; Through his father, Kiedis, who worked under the stage name of Cole Dammett, landed his first acting role, appearing as [[Sylvester Stallone]]'s character's son in the 1978 film, ''[[F.I.S.T. (film)|F.I.S.T.]]'' He landed two more acting jobs, in an [[ABC Afterschool Special]] and an appearance in the film ''Jokes My Folks Never Told Me''.{{citation needed|date=September 2020}}

394 It was initially decided to first sign an [[A-list]] actor for Superman before Richard Donner was hired as director. [[Robert Redford]] was offered a large sum, but felt he was too famous. [[Burt Reynolds]] also turned down the role, while [[Sylvester Stallone]] was interested and met with Donner, but he was more interested casting an &quot;unknown&quot; actor. [[Paul Newman]] was offered his choice of roles as Superman, Lex Luthor or Jor-El for $4&amp;nbsp;million, turning down all three roles.&lt;ref name=&quot;Ilya&quot; /&gt;&lt;ref&gt;{{Cite web |date=April 2016 |title='Superman,' the Inside Story: Director Richard Donner Remembers Meeting Stallone to Play the Lead, Working with Brando, and a Near-Fatal Knife Attack |url=https://www.hollywoodreporter.com/movies/movie-features/superman-inside-story-director-richard-879894/ |url-status=live |archive-url=https://web.archive.org/web/20231202051501/https://www.hollywoodreporter.com/movies/movie-features/superman-



```
    inside-story-director-richard-879894/ |archive-date=December 2, 2023 |access-date=December 2, 2023 |website=[[The Hollywood Reporter]]}}&lt;/ref&gt;
395 In 1984, Del Wood appeared in the movie [[Rhinestone (film)|''Rhinestone'']], starring [[Dolly Parton]] and [[Sylvester Stallone]] as the pianist in the Wild Possums Band.
396 Aspel was the host of the chat show ''Aspel &amp; Company'', which ran from 9 June 1984 to 20 June 1993 on [[ITV (TV network)|ITV]]. The show was successful in attracting high-profile guests
    including then-Prime Minister [[Margaret Thatcher]], [[George Harrison]] and [[Ringo Starr]]. In 1993, ''Aspel &amp; Company'' was censured by the [[Independent Television Commission]] because of an
    interview with [[Arnold Schwarzenegger]], [[Bruce Willis]] and [[Sylvester Stallone]] who were promoting their joint business venture [[Planet Hollywood]]. For a time, ''Aspel &amp; Company''
    performed well for ITV in the highly competitive Saturday night ratings but after the Planet Hollywood controversy, Aspel vowed never to host a chat show again.&lt;ref name=&quot;Inde&quot;&gt;{{cite
    news|author=Adrian Turpin|title=and my next guest is... |url=https://www.independent.co.uk/news/media/and-my-next-guest-istony-blair--stars-on-the-couch-468523.html
    |archive-url=https://ghostarchive.org/archive/20220512/https://www.independent.co.uk/news/media/and-my-next-guest-istony-blair--stars-on-the-couch-468523.html |archive-date=12 May 2022 |url-
    access=subscription |url-status=live |date=4 March 2006|access-date=27 December 2009 |work=The Independent |location=London}}&lt;/ref&gt;
397 Her first feature film was 1972 comedy ''[[The Honkers]]'' co-starring opposite [[James Coburn]] and [[Lois Nettleton]]. She later had supporting roles in ''[[Cancel My Reservation]]'' (1972),
    ''[[The All-American Boy (film)|The All-American Boy]]'' (1973), and ''[[Trackdown (film)|Trackdown]]'' (1976). In 1976, she had a female leading role in the drama film ''[[Lifeguard (film)|
    Lifeguard]]'' starring alongside [[Sam Elliott]]. She auditioned for the role of [[Lois Lane]] in the 1978 superhero film ''[[Superman (1978 film)|Superman]]'', a role eventually awarded to [[Margot
    Kidder]].&lt;ref name=&quot;tvg&quot;/&gt; Archer continued to appear in feature films, including ''[[Good Guys Wear Black]]'' (1978) starring [[Chuck Norris]], ''[[Paradise Alley]]'' (1978) opposite
    [[Sylvester Stallone]], and ''[[Hero at Large]]'' (1980), co-starring [[John Ritter]].
398 | &lt;ref&gt;{{cite web|url=https://deadline.com/2021/10/sylvester-stallone-announces-exit-from-the-expendables-franchise-1234857872/|title=Sylvester Stallone Announces Exit From 'Expendables'
    Franchise As He Wraps Production On Fourth Installment|website=Deadline Hollywood|first=Matt|last=Grobar|date=October 18, 2021|access-date=October 22, 2021}}&lt;/ref&gt;
399 Stowe postponed her acting career in 1996 to concentrate on her family life. She settled for several years in a Texas ranch, with her daughter May and husband [[Brian Benben]].&lt;ref
    name=&quot;people&quot;&gt;[http://people.com/archive/madeleine-stowe-best-revenge-vol-76-no-24/ Madeleine Stowe: Best Revenge], ''[[People (magazine)|People]]'', December 12, 2011.&lt;/ref&gt; In
    1998, she came back with ''[[The Proposition (1998 film)|The Proposition]]'' and ''[[Playing by Heart]]'', and then ''[[The General's Daughter (film)|The General's Daughter]]'', opposite [[John
    Travolta]] in 1999.&lt;ref name=yahoo&gt;{{cite web|title=Madeleine Stowe-
    Biography|url=https://movies.yahoo.com/person/madeleine-stowe/biography.html;_ylt=AqmMAJIE6ppxSBCjk.9ZRYj3TssF;_ylu=X3oDMTI1Zmlwbjk5BG1pdANQZXJzb24gSGVhZGVyIE1TQgRwb3MDMwRzZWMDTWVkaWFFbnRpdHlIZWFkZXJ
    MaW5rc1BhY2thZ2VBc3NlbWJseQ--;_ylg=X3oDMTFjcjk4NTY4BGludGwDdXMEbGFuZwNlbi11cwRwc3RhaWQDBHBzdGNhdAMEcHQDBHRlc3QD;_ylv=3|publisher=[[Yahoo!]]|access-date=February 17, 2011}}&lt;/ref&gt; In 2001, she
    starred in the science-fiction [[box office bomb]] ''[[Impostor (2002 film)|Impostor]]''.&lt;ref&gt;{{cite web|url=https://www.boxofficemojo.com/movies/?id=impostor.htm|title=Impostor -
    boxofficemojo|access-date=November 28, 2014}}&lt;/ref&gt; In 2002, she played [[Julia Compton Moore|Julia Moore]] in the war film ''[[We Were Soldiers]]'' with [[Mel Gibson]], and the box office flop
    action-comedy ''[[Avenging Angelo]]'' opposite [[Sylvester Stallone]].&lt;ref name=yahoo/&gt; In 2003, she starred in the thriller ''[[Octane (film)|Octane]]'' as Senga Wilson, a single mother trying
    to save her teenage daughter ([[Mischa Barton]]) from a bizarre cult obsessed with blood and cars.
400 The knife is worn on a pistol belt on the bear's left side with the Army Combat Uniform. The Arkansas Brigade Bowie knife has been worn by members through two deployments in support of Operation
    Iraqi Freedom II. The knife continues to be produced in Arkansas. Until his death, each presentation-grade knife was handmade by Mr. Jimmy Lile of Russellville, Arkansas. Mr. Lile was also
    commissioned to make the knives made by Sylvester Stallone in the &quot;Rambo&quot; movies. The Lile family continued to make the Bowie knife for the unit for several years following Mr. Lile's
    death. Today the brigade combat team's knife is produced by Mr. Kenny Teague of [[Mountainburg, Arkansas|Mountainburg]]. The general public cannot purchase one of these knives, but can purchase a
    different style based on the Bowie knife pattern. Each brigade Bowie knife bears the recipient's name, social security number, rank, and military branch, as well as the maker's name and serial number
    of the knife.&lt;ref name=&quot;Infantry Brigade 1997&quot;/&gt;
401 A number of Ipswich players appeared alongside [[Sylvester Stallone]] and [[Pelé]] in the 1981 [[prisoner of war]] film ''[[Escape to Victory]]'', including [[John Wark]], [[Russell Osman]], [[Robin
    Turner (footballer)|Robin Turner]], [[Laurie Sivell]], and [[Kevin O'Callaghan]]. Other Ipswich Town players stood in for actors in the football scenes-[[Kevin Beattie]] for [[Michael Caine]], and
    [[Paul Cooper (footballer, born 1953)|Paul Cooper]] for Sylvester Stallone.&lt;ref&gt;{{cite web | url = http://www.eadt.co.uk/sport/town_players_in_escape_to_victory_reunion_1_835075 | work = East
    Anglian Daily Times | title = Town players in 'Escape to Victory' reunion | first = Elvin | last = King | date = 23 March 2011 | access-date = 16 April 2016 | archive-date = 23 January 2017 |
    archive-url = https://web.archive.org/web/20170123075114/http://www.eadt.co.uk/sport/town_players_in_escape_to_victory_reunion_1_835075 | url-status = dead }}&lt;/ref&gt;
402 * ''[[First Blood]]'' (1985 NTV edition) ([[John Rambo]] ([[Sylvester Stallone]]))&lt;ref&gt;{{cite web|url= https://www.hmv.co.jp/artist_%E3%83%A9%E3%83%B3%E3%83%9C%E3%83%BC_000000000101734/item_
    %E3%83%A9%E3%83%B3%E3%83%9C%E3%83%BC-4K%E3%83%AC%E3%82%B9%E3%83%88%E3%82%A2%E7%89%88_10419173|title= ランボー4K レストア版|access-date= October 5, 2021|work= HMV|archive-date= October 5, 2021|
    archive-url= https://web.archive.org/web/20211005222522/https://www.hmv.co.jp/artist_%E3%83%A9%E3%83%B3%E3%83%9C%E3%83%BC_000000000101734/item_%E3%83%A9%E3%83%B3%E3%83%9C%E3%83%BC-4K%E3%83%AC
    %E3%82%B9%E3%83%88%E3%82%A2%E7%89%88_10419173|url-status= live}}&lt;/ref&gt;
403 *[[Sylvester Stallone]] (born 1946), American actor
404     <title>Silvester Stallone</title>
405     <redirect title="Sylvester Stallone" />
406       <text bytes="32" sha1="bnfj12dj30potn771e5t1yex066t6ak" xml:space="preserve">#redirect [[Sylvester Stallone]]</text>
407 Promoted to president, then to chairman of the board, Ovitz's roles at CAA were numerous. He served as talent agent to Hollywood actors [[Tom Cruise]], [[Dustin Hoffman]], [[Kevin Costner]], [[John
    Belushi]], [[Michael Douglas]], [[Bill Murray]], [[Sylvester Stallone]], and [[Barbra Streisand]], as well as directors [[Steven Spielberg]], [[Barry Levinson]], and [[Sydney Pollack]].&lt;ref&gt;
    {{Citation | title = Michael Ovitz: A Tough, Innovative Superagent Emerges as King of the Hollywood Deal | newspaper=[[People (magazine)|People]] | date = Dec 31, 1990 | url =
    http://www.people.com/people/archive/article/0,,20114127,00.html}}&lt;/ref&gt; He also provided corporate consulting services, helping negotiate several major international business mergers and deals
    including [[Panasonic|Matsushita]]'s acquisition of [[MCA Inc.|MCA]]/[[Universal Pictures|Universal]], the financial rescue of [[Metro-Goldwyn-Mayer|MGM]]/[[United Artists]], and [[Sony]]'s
    acquisition of [[Columbia Pictures]].&lt;ref&gt;{{Citation | last = Castro | first = Janice | title = In A Rare Interview, Ovitz Defends His Power | magazine= Time |  date = Apr 19, 1993 | url
    =http://www.time.com/time/magazine/article/0,9171,978256-1,00.html | archive-url =https://web.archive.org/web/20121027013409/http://www.time.com/time/magazine/article/0,9171,978256-1,00.html | url-
    status =dead | archive-date =October 27, 2012 }}&lt;/ref&gt; Ovitz's signing of Coca-Cola as a CAA client from agency McCann-Erickson had a significant impact on the advertising industry.&lt;ref&gt;
    {{Citation | title = Top 100 People of the Advertising Century  | url =http://adage.com/century/people087.html }}&lt;/ref&gt; He negotiated [[David Letterman]]'s move from [[NBC]] to [[CBS]],
    chronicled in the book ''[[The Late Shift (book)|The Late Shift: Letterman, Leno, and the Network Battle for the Night]]'' by Bill Carter.&lt;ref&gt;{{Cite news| url =
    https://www.nytimes.com/1994/01/30/magazine/behind-the-headlines-in-the-leno-letterman-war.html |title=Behind the Headlines in the Leno -- Letterman War|author=Bill Carter|newspaper=[[The New York
    Times]] |date =January 30, 1994}}&lt;/ref&gt; He disliked publicity, however, with the ''Times'' reporting that &quot;Ovitz is one of the very few people in the world who own almost all the
    photographs ever taken of them&quot;.{{r|davis19890709}}
408 [[David Zucker (filmmaker)|David Zucker]], who was a fan of ''Cannibal!'', contacted the duo to produce a 15-minute short film for [[Seagram]] to show at a party for its acquisition of [[Universal
    Pictures|Universal Studios]].&lt;ref name=&quot;yourstudio&quot;/&gt; Due to a misunderstanding, Parker and Stone [[improvisation|improvised]] much of the film an hour before it was shot, creating it
    as a spoof of 1950s instructional videos.&lt;ref name=&quot;yourstudio&quot;/&gt; The result, ''[[Your Studio and You]]'', features numerous celebrities, including [[Sylvester Stallone]], [[Demi
    Moore]], and [[Steven Spielberg]]. &quot;You could probably make a feature film out of the experience of making that movie because it was just two dudes from college suddenly directing Steven
    Spielberg&quot;, Parker later remarked, noting that the experience was difficult for the two.&lt;ref name=&quot;yourstudio&quot;&gt;{{cite news|
    url=http://treyparker.info/archives_zap2it_16jul01.htm| title='South Park' Creator Trey Parker Cops to Kooky Universal Spoof| date=July 16, 2001| publisher=[[Zap2it]]| access-date=July 2, 2014|
    first=Stephen| last=Galloway| url-status=dead| archive-url=https://web.archive.org/web/20141028085139/http://treyparker.info/archives_zap2it_16jul01.htm| archive-date=October 28, 2014}}&lt;/ref&gt;
409 [[David Zucker (filmmaker)|David Zucker]], who was a fan of ''Cannibal!'', contacted the duo to produce a 15-minute short film for [[Seagram]] to show at a party for its acquisition of [[Universal
    Pictures|Universal Studios]].&lt;ref name=&quot;yourstudio&quot;/&gt; Due to a misunderstanding, Parker and Stone [[improvisation|improvised]] much of the film an hour before it was shot, creating it
    as a spoof of 1950s instructional videos.&lt;ref name=&quot;yourstudio&quot;/&gt; The result, ''[[Your Studio and You]]'', features numerous celebrities, including [[Sylvester Stallone]], [[Demi
    Moore]], and [[Steven Spielberg]]. &quot;You could probably make a feature film out of the experience of making that movie because it was just two dudes from college suddenly directing Steven
    Spielberg,&quot; Parker later remarked, noting that the experience was difficult for the two.&lt;ref name=&quot;yourstudio&quot;&gt;{{cite news|
    url=http://treyparker.info/archives_zap2it_16jul01.htm| title='South Park' Creator Trey Parker Cops to Kooky Universal Spoof| date=July 16, 2001| publisher=[[Zap2it]]| access-date=July 2, 2014|
    first=Stephen| last=Galloway| url-status=dead| archive-url=https://web.archive.org/web/20141028085139/http://treyparker.info/archives_zap2it_16jul01.htm| archive-date=October 28, 2014| df=mdy-
    all}}&lt;/ref&gt;
410 [[Yashpal Sharma (actor)|Yashpal Sharma]] was chosen by Gowariker for the role of Lakha, the woodcutter, after his portrayal in ''[[Samar (1999 film)|Samar]]'' (1999). He said it was a good
    experience working with Khan and Gowariker during the film.&lt;ref&gt;{{cite web|url=http://www.rediff.com/movies/2001/jun/27lag6.htm |title=Lakha, the woodcutter: The Ten Samurai |work=Rediff.com |
    access-date=29 December 2007 |url-status=live |archive-url=https://web.archive.org/web/20040826000951/http://www.rediff.com/movies/2001/jun/27lag6.htm |archive-date=26 August 2004 }}&lt;/ref&gt; Amin
    Hajee earlier worked in a film with Gowariker. The friendly association brought Gowariker to him with the script, which he liked, and thereafter he successfully auditioned for his role of Bagha, a
    mute drummer. His knowledge of mute people and some assistance from a music band helped him better prepare for his role. Gowariker, who believed that Amin was like [[Sylvester Stallone]], would refer
    to him as Stallone during filming.&lt;ref name=&quot;bagha_intw&quot;&gt;{{cite web|url=http://www.rediff.com/movies/2001/jun/27lag1.htm |title=Bagha, the mute: The Ten Samurai |work=Rediff.com |
    access-date=29 December 2007 |url-status=live |archive-url=https://web.archive.org/web/20050913005125/http://rediff.com/movies/2001/jun/27lag1.htm |archive-date=13 September 2005 }}&lt;/ref&gt;
    [[Aditya Lakhia]]'s association with Gowariker in ''[[Kabhi Haan Kabhi Naa]]'' (1993) and ''Pehla Nasha'' (1993) brought him the role of Kachra, the untouchable. He read the book ''[[Everybody Loves
    a Good Drought]]'' by [[Palagummi Sainath|P. Sainath]] to better understand and portray his character.&lt;ref name=&quot;kachra_intw&quot;&gt;{{cite
    web|url=http://www.rediff.com/movies/2001/jun/27lag2.htm |title=Kachra, the untouchable: The Ten Samurai |work=Rediff.com |access-date=29 December 2007 |url-status=live
    |archive-url=https://web.archive.org/web/20050419213010/http://rediff.com/movies/2001/jun/27lag2.htm |archive-date=19 April 2005 }}&lt;/ref&gt;
411 *[[Sylvester Stallone filmography|Sylvester Stallone]]
412 |title=Henry Winkler Accidentally Helped Sylvester Stallone Make Rocky
413 Several years after the release of the movie, [[Sylvester Stallone]] claimed that he turned down the role of Louis Gara.&lt;ref&gt;{{Cite web |url=http://www.macleans.ca/general/on-feuding-with-
    arnold-schwarzenegger-and-whats-wrong-with-todays-action-heroes/ |title=In conversation: Sylvester Stallone |date=March 6, 2012|access-date = February 24, 2016 |website=MacLean's |last=Johnson |
```

first=Brian D.|archive-date = October 5, 2020|archive-url = https://web.archive.org/web/20201005150124/https://www.macleans.ca/general/on-feuding-with-arnold-schwarzenegger-and-whats-wrong-with-todays-action-heroes/|url-status = live}}&lt;/ref&gt; Tarantino considered [[Paul Newman]], [[Gene Hackman]] and [[John Saxon]] for the role of Max Cherry, before casting [[Robert Forster]].&lt;ref&gt;{{Cite web|url = http://wearemoviegeeks.com/2009/08/tarantino-week-revisiting-jackie-brown/|title = Tarantino Week: Revisiting 'Jackie Brown'|date = August 19, 2009|access-date = August 5, 2021|archive-date = August 5, 2021|archive-url = https://web.archive.org/web/20210805185207/http://wearemoviegeeks.com/2009/08/tarantino-week-revisiting-jackie-brown/|url-status = live}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://theplaylist.net/quentin-tarantino-newman-hackman-jackie-brown-20200203/|title=Quentin Tarantino Says He Originally Considered Paul Newman &amp; Gene Hackman for the Max Cherry Role in 'Jackie Brown'|access-date=August 5, 2021|archive-date=August 5, 2021|archive-url=https://web.archive.org/web/20210805185659/https://theplaylist.net/quentin-tarantino-newman-hackman-jackie-brown-20200203/|url-status=live}}&lt;/ref&gt;

414 While auditioning for the role of [[Apollo Creed]] alongside [[Sylvester Stallone]] in ''[[Rocky]]'', Weathers criticized Stallone's acting, which led to him getting the role.&lt;ref&gt;{{cite news |last=Parker |first=Ryan |date=December 3, 2015 |title=Carl Weathers Looks Back on Creed: Sparring With Ali, Slighting Stallone and Passing the Torch |url=https://www.hollywoodreporter.com/news/carl-weathers-looks-back-creed-845151 |work=[[The Hollywood Reporter]] |access-date=July 22, 2017 |archive-date=July 2, 2017 |archive-url=https://web.archive.org/web/20170702072736/http://www.hollywoodreporter.com/news/carl-weathers-looks-back-creed-845151 |url-status=live }}&lt;/ref&gt; He reprised the role of Apollo Creed in the next three ''Rocky'' films: ''[[Rocky II]]'' (1979), ''[[Rocky III]]'' (1982), and ''[[Rocky IV]]'' (1985).&lt;ref&gt;{{Cite web|url=https://ew.com/movies/2017/04/27/carl-weathers-rocky-love-story/|title=Carl Weathers Debates Whether 'Rocky' is a Sports Movie or a Love Story|website=EW.com|access-date=February 2, 2024|archive-date=August 17, 2023|archive-url=https://web.archive.org/web/20230817204201/https://ew.com/movies/2017/04/27/carl-weathers-rocky-love-story/|url-status=live}}&lt;/ref&gt;

415 For the sixth film in the ''Rocky'' series, ''[[Rocky Balboa (film)|Rocky Balboa]]'' (2006), Stallone asked Weathers, [[Mr. T]], and [[Dolph Lundgren]] for permission to use footage from their appearances in the earlier ''Rocky'' films. Mr. T and Lundgren agreed, but Weathers wanted an actual part in the movie, even though his character had died in ''[[Rocky IV]]''.&lt;ref name=&quot;inquisitr&quot;&gt;{{cite news |last=Mitchell |first=Aric |date=November 24, 2015 |title=Carl Weathers, Sylvester Stallone May Have Mended Their Differences Over 'Creed' |url=https://www.inquisitr.com/2587362/carl-weathers-sylvester-stallone-may-have-mended-their-differences-over-creed/ |work=[[Inquisitr]] |access-date=September 23, 2016 |archive-date=April 14, 2018 |archive-url=https://web.archive.org/web/20180414010616/https://www.inquisitr.com/2587362/carl-weathers-sylvester-stallone-may-have-mended-their-differences-over-creed/ |url-status=live }}&lt;/ref&gt; Stallone refused, and Weathers decided not to allow Stallone to use his image for flashbacks from the previous films. They instead used footage of a fighter who looks similar to Weathers.&lt;ref&gt;stated by Stallone on ''[[Jimmy Kimmel Live!]]'' (December 21, 2006).&lt;/ref&gt; Weathers and Stallone patched up their differences and Weathers agreed to allow footage of him from previous films to be used throughout ''[[Creed (film)|Creed]]'' (2015).&lt;ref name=&quot;inquisitr&quot;/&gt;

416 [[Sylvester Stallone]] had initially agreed to a cameo appearance in the final act, where Newman imagines himself as a [[John Rambo]]-type soldier on a mission to rescue Stanley Spadowski, but Stallone ultimately declined.&lt;ref&gt;{{cite web|url=https://screenrant.com/uhf-movie-sylvester-stallone-rambo-cameo-rejection/|publisher=[[Screen Rant]]|last=Cotter | first=Padraig|title=Why Stallone Backed Out Of A Cameo In UHF's Rambo Parody Sequence|language=en-US|url-status=live|date=February 19, 2022|accessdate=March 26, 2022|archivedate=February 19, 2022|archiveurl=https://web.archive.org/web/20220219194622/https://screenrant.com/uhf-movie-sylvester-stallone-rambo-cameo-rejection/}}&lt;/ref&gt;

417 It was announced in June 2010 that he would be joining the cast of the [[CBS]] soap opera ''[[The Young and the Restless]]'' starting July 12.&lt;ref&gt;{{cite web|url=https://www.tvguide.com/news/eric-roberts-restless-1019721/|title=Eric Roberts Joining the Cast of The Young and the Restless|publisher=TVGuide.com|first=Kate|last=Stanhope|date=June 17, 2010|access-date=January 1, 2022}}&lt;/ref&gt; The following month saw the release of the action film ''[[The Expendables (2010 film)|The Expendables]]'' in which Roberts plays a lead villain. It was directed by and starred [[Sylvester Stallone]], with [[Jason Statham]], [[Jet Li]], [[Steve Austin]], [[Gary Daniels]], [[Dolph Lundgren]], [[Randy Couture]], [[Terry Crews]], [[David Zayas]], and [[Mickey Rourke]]. The film is about a group of elite mercenaries called The Expendables (Stallone, Statham, Li, Lundgren, Couture, and Crews) who are on a mission to overthrow dictator General Garza (Zayas) in Vilena, an island in the [[Gulf of Mexico]]. It is revealed that an ex-[[Central Intelligence Agency|CIA]] officer James Munroe (Roberts) is keeping Garza in power as a [[figurehead]] for his own profiteering operations. With his two deadly bodyguards Dan Paine (Austin) and The Brit (Daniels), they become a major obstacle in the way of The Expendables.&lt;ref&gt;{{cite AV media|url=https://www.filmo.gs/release/23737-the-expendables|title=The Expendables |date=2010 |type=DVD |language=en, es |publisher=Lionsgate, Millenium, and Nu Image|id=0 31398 12842 7}}&lt;/ref&gt; Later that year, he appeared with [[Stone Cold Steve Austin|Steve Austin]] and [[Gary Daniels]], his co-stars from ''[[The Expendables (2010 film)|The Expendables]]'', in the 2010 action film ''[[Hunt to Kill]]''. In October 2010, he played the major role in the American family movie ''First Dog'', which received positive reviews.&lt;ref&gt;{{Cite web|url=https://www.rottentomatoes.com/m/first_dog|title=First Dog (2010)|via=www.rottentomatoes.com}}&lt;/ref&gt; December 2010 saw the premiere of the [[List of Celebrity Rehab with Dr. Drew episodes#Season 4|fourth season]] of ''[[Celebrity Rehab with Dr. Drew]]'', which documented Roberts' struggle with dependency on [[medical marijuana]].&lt;ref name=&quot;Huff&quot;&gt;{{cite web |last1=Hall |first1=Katy |title=Eric Roberts' Celebrity Rehab Addiction: Medical Marijuana |url=https://www.huffpost.com/entry/eric-roberts-celebrity-re_n_659961 |website=HuffPost |access-date=January 1, 2022 |language=en |date=September 25, 2010}}&lt;/ref&gt; His wife [[Eliza Roberts (actress)|Eliza]] and his stepson [[Keaton Simons]] appeared in episode 6 to discuss the effects of his addiction on their lives.&lt;ref&gt;''[[Celebrity Rehab with Dr. Drew]]'': Episode 4.6, [[VH1]] (premiered December 22, 2010).&lt;/ref&gt;

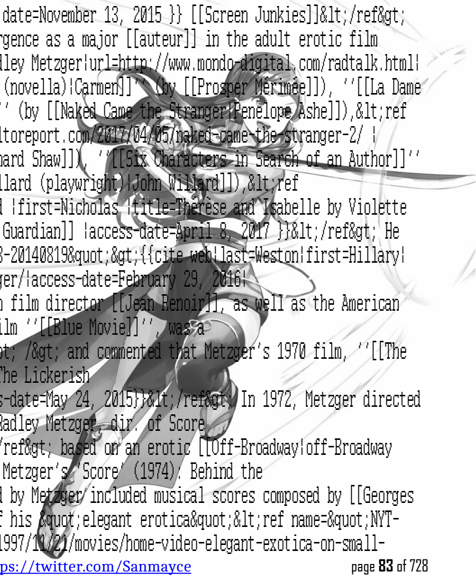
418 Jones played a professional killer in the Kazakhstani film ''Liquidator'' in 2011.&lt;ref name=&quot;TV&quot;/&gt; His character is an elite assassin invited to eliminate the main character. Producers of the film dealt with the Kazakh-to-English language barrier by writing Jones' character as a mute who does not speak. In the same year, he played Zed in the movie ''Blood Out''. He played a role in the Hungarian film ''The Magic Boys'' in 2012.&lt;ref name=&quot;TV&quot;/&gt; That same year, he voiced Freddie the Dog in ''[[Madagascar 3: Europe's Most Wanted]]''.&lt;ref name=&quot;TV&quot;/&gt; He co-starred alongside [[Sylvester Stallone]] and [[Arnold Schwarzenegger]] in the action-thriller ''[[Escape Plan (film)|Escape Plan]]'',&lt;ref name=&quot;TV&quot;/&gt; released in 2013, and was featured with [[Danny Trejo]] in the 2014 horror-thriller ''[[Reaper (film)|Reaper]]''.&lt;ref name=&quot;TV&quot;/&gt;

419 [[Principal photography]] began in the fall of 2002,&lt;ref name=&quot;ESPN&quot;/&gt; and [[Matthew F. Leonetti]] served as the [[director of photography]].&lt;ref&gt;{{cite news|title=2 Fast 2 Furious Production Notes - The Cars|url=http://www.cinemareview.com/production.asp?prodid=2113#|access-date=July 21, 2017|work=Cinemareview.com|publisher=Universal Studios|archive-url=https://web.archive.org/web/20170729133550/http://www.cinemareview.com/production.asp?prodid=2113|archive-date=July 29, 2017|url-status=dead}}&lt;/ref&gt; Filming was done mostly in various parts of [[South Florida]] such as [[Miami Beach, Florida|Miami Beach]], [[Seven Mile Bridge]], and Homestead Air Reserve Base.&lt;ref name=&quot;prodnotes&quot;/&gt;&lt;ref&gt;{{cite news|last1=Miller|first1=Michael E.|title=Best and Worst Movies Shot in Miami Beach, From ''Scarface'' to Sly Stallone's ''The Specialist''|url=http://www.miaminewtimes.com/news/best-and-worst-movies-shot-in-miami-beach-from-scarface-to-sly-stallones-the-specialist-6553280|access-date=July 21, 2017|work=[[Miami New Times]]|publisher=Voice Media Group|date=November 16, 2012|archive-date=July 29, 2017|archive-url=https://web.archive.org/web/20170729133626/http://www.miaminewtimes.com/news/best-and-worst-movies-shot-in-miami-beach-from-scarface-to-sly-stallones-the-specialist-6553280|url-status=live}}&lt;/ref&gt; Hauser's character's mansion was shot in [[Coral Gables, Florida|Coral Gables]], in a house owned by [[Sylvester Stallone]].&lt;ref name=&quot;prodnotes&quot;/&gt; At [[Bill Baggs Cape Florida State Park]], ''2 Fast 2 Furious'' was filmed on one side, while ''[[Bad Boys II]]'' was filmed on the other side at the same time.&lt;ref&gt;{{cite web|url=https://bashful269.wordpress.com/2020/01/24/things-you-never-noticed-in-4-bad-boys-ii/|title=Things You Never Noticed In #4: Bad Boys II|date=January 24, 2020 |access-date=May 16, 2023|archive-date=May 16, 2023|archive-url=https://web.archive.org/web/20230516051207/https://bashful269.wordpress.com/2020/01/24/things-you-never-noticed-in-4-bad-boys-ii/|url-status=live}}&lt;/ref&gt;{{Better source needed|date=May 2023}}

420 * The 2001 film ''[[Driven (2001 film)|Driven]]'' starring [[Sylvester Stallone]] and [[Burt Reynolds]] was partially filmed in Uxbridge.

421 Besides being known for its architecture and collections, the Philadelphia Museum of Art has in recent decades become known due to the role it played in the [[Rocky (franchise)#Films|''Rocky'' films]]{{emdash}}''[[Rocky]]'' (1976) and seven of its eight sequels, ''[[Rocky II|II]]'', ''[[Rocky III|III]]'', ''[[Rocky V|V]]'', ''[[Rocky Balboa (film)|Rocky Balboa]]'', ''[[Creed (film)|Creed]]'', ''[[Creed II]]'', and ''[[Creed III]]''. Visitors to the museum are often seen mimicking [[Rocky Balboa]]'s (portrayed by [[Sylvester Stallone]]) famous run up the east entrance stairs, informally nicknamed the [[Rocky Steps]].&lt;ref&gt;[http://www.visitphilly.com/museums-attractions/philadelphia/the-rocky-statue-and-the-rocky-steps/ The Rocky Statue and the Rocky Steps] {{Webarchive|url=https://web.archive.org/web/20180309104134/http://www.visitphilly.com/museums-attractions/philadelphia/the-rocky-statue-and-the-rocky-steps/ |date=March 9, 2018 }} visitphilly.com, accessed June 17, 2011.&lt;/ref&gt; [[Screen Junkies]] named the museum's stairs the second most famous movie location behind only [[Grand Central Terminal|Grand Central Station]] in New York.&lt;ref&gt;[http://www.screenjunkies.com/movies/movie-lists/10-most-famous-movie-locations/ ''10 Most Famous Movie Locations''] {{Webarchive|url=https://web.archive.org/web/20151113074139/http://www.screenjunkies.com/movies/movie-lists/10-most-famous-movie-locations/ |date=November 13, 2015 }} [[Screen Junkies]]&lt;/ref&gt;

422 Metzger's second directorial effort, ''[[The Dirty Girls (1965 film)|The Dirty Girls]]'' (shot in 1963 and released in 1965), marked his emergence as a major [[auteur]] in the adult erotic film genre. His subsequent films were often shot in Europe&lt;ref name=&quot;mondointerview&quot;&gt;{{cite web|author=Staff|title=A Talk With Radley Metzger|url=http://www.mondo-digital.com/radtalk.html|access-date=May 23, 2015|website=Mondo-Digital.com}}&lt;/ref&gt; and adapted from [[novel]]s or other literary sources, including ''[[Carmen (novella)|Carmen]]'' (by [[Prosper Mérimée]]), ''[[La Dame aux Camélias]]'' (by [[Alexandre Dumas]]), ''[[The Image (novel)|L'image]]'' (by [[Catherine Robbe-Grillet]]), ''[[Naked Came the Stranger]]'' (by [[Naked Came the Stranger|Penelope Ashe]]),&lt;ref name=&quot;TRR-20170405&quot;&gt;{{cite web |author=Staff |title='Naked Came the Stranger' (1975): The Hoax, The Film |url=http://www.therialtoreport.com/2017/04/05/naked-came-the-stranger-2/ |date=April 5, 2017 |work=[[The Rialto Report]] |access-date=April 5, 2017 }}&lt;/ref&gt; ''[[Pygmalion (play)|Pygmalion]]'' (by [[George Bernard Shaw]]), ''[[Six Characters in Search of an Author]]'' (by [[Luigi Pirandello]]),&lt;ref name=&quot;TCM-2015&quot; /&gt; ''[[The Cat and the Canary (play)|The Cat and the Canary]]'' (by [[John Willard (playwright)|John Willard]]),&lt;ref name=mondointerview/&gt; and ''Thérèse et Isabelle'' (by [[Violette Leduc]]).&lt;ref name=&quot;TG-20120228&quot;&gt;{{cite news |last=Lezard |first=Nicholas |title=Thérèse and Isabelle by Violette Leduc - review |url=https://www.theguardian.com/books/2012/feb/28/therese-isabelle-violette-leduc-review |date=February 28, 2012 |work=[[The Guardian]] |access-date=April 8, 2017 }}&lt;/ref&gt; He cites [[John Farrow]], [[Claude Lelouch]],&lt;ref name=&quot;FSLC-2014&quot; /&gt; [[Michael Powell]], [[Alain Resnais]]&lt;ref name=&quot;BB-20140819&quot;&gt;{{cite web|last=Weston|first=Hillary|date=August 19, 2014|title=Porn Before It Was Chic: An Interview With Radley Metzger on Sex and Cinema|url=http://bbook.com/film/radley-metzger/|access-date=February 29, 2016|work=[[BlackBook]]}}&lt;/ref&gt; and [[Orson Welles]] as influencing his work.&lt;ref name=mondointerview/&gt; Metzger worked with the French film director [[Jean Renoir]], as well as the American actor [[Hal Linden]].&lt;ref name=&quot;FM-20140807&quot; /&gt; [[Andy Warhol]], who helped begin the [[Golden Age of Porn]] with his 1969 film ''[[Blue Movie]]'', was a [http://www.therialtoreport.com/wp-content/uploads/2017/04/RR_Radley_Warhol.jpg fan] of Metzger's film work&lt;ref name=&quot;FM-20140807&quot; /&gt; and commented that Metzger's 1970 film, ''[[The Lickerish Quartet]]'', was "an outrageously kinky masterpiece".&lt;ref name=&quot;FSLC-AndyWarhol&quot;&gt;{{cite web|author=Staff |title=''The Lickerish Quartet''|url=http://www.filmlinc.com/films/on-sale/the-lickerish-quartet |date=August 8, 2014|work=[[Film Society of Lincoln Center]]|access-date=May 24, 2015}}&lt;/ref&gt; In 1972, Metzger directed the film ''[[Score (1974 film)|Score]]'',&lt;ref name=&quot;CINE-20140108&quot;&gt;{{cite web |last=Kramer |first=Gary M. |title=Interview: Radley Metzger, dir. of Score |url=http://cinedelphia.com/interview-radley-metzger-dir-of-score/ |date=January 8, 2014 |work=Cinedelphia |access-date=April 4, 2017 }}&lt;/ref&gt; based on an erotic [[Off-Broadway|off-Broadway play]] that included [[Sylvester Stallone]].&lt;ref name=&quot;TRR-20170404&quot;&gt;{{cite web|author=Staff|date=April 4, 2017|title=Radley Metzger's 'Score' (1974): Behind the Scenes|url=http://www.therialtoreport.com/2017/04/04/score/|access-date=April 4, 2017|work=[[The Rialto Report]]}}&lt;/ref&gt; Films directed by Metzger included musical scores composed by [[Georges Auric]], [[Stelvio Cipriani]], [[Georges Delerue]], and [[Piero Piccioni]].&lt;ref name=mondointerview/&gt; Metzger's signature film style of his &quot;elegant erotica&quot;&lt;ref name=&quot;NYT-19971121&quot;&gt;{{cite news |last=Nichols |first=Peter M. |title=Home Video; Elegant Exotica On Small Screen |url=https://www.nytimes.com/1997/11/21/movies/home-video-elegant-exotica-on-small-



screen.html |date=November 21, 1997 |work=[[New York Times]] |access-date=April 13, 2017 }}&lt;/ref&gt; had developed into being &quot;a Euro-centric combination of stylish decadence, wealth and the aristocratic&quot;.&lt;ref name=&quot;TRR-20170403&quot;/&gt;

423 The success of ''Blair Witch'' as an indie project remains among the few exceptions, however, and control of [[Big Five (movie studios)|The Big Five]] studios over film{{nbsp}}making continued to increase through the 1990s. The Big Six companies all enjoyed a period of expansion in the 1990s. They each developed different ways to adjust to rising costs in the film industry, especially the rising salaries of movie stars, driven by powerful agents. The biggest stars like [[Sylvester Stallone]], [[Russell Crowe]], [[Tom Cruise]], [[Nicole Kidman]], [[Sandra Bullock]], [[Arnold Schwarzenegger]], [[Mel Gibson]], [[Kevin Bacon]], and [[Julia Roberts]] received between $15-$20&amp;nbsp;million per film and in some cases were even given a share of the film's profits.&lt;ref name=cin08 /&gt;

424 |A film adaptation of [[Don Pendleton]]'s [[The Executioner (book series)|''The Executioner'' series]] written by [[Hilary Henkin]] starring [[Sylvester Stallone]] and [[Cynthia Rothrock]]

425 |''The Gambler'', a film written by [[Edward Neumeier]] and Michael Miner starring [[Sylvester Stallone]]

426 '''''Rambo''''' is an American [[media franchise]] centered on a series of [[action film]]s featuring [[John Rambo|John J. Rambo]]. The five films are ''[[First Blood]]'' (1982), ''[[Rambo: First Blood Part II]]'' (1985), ''[[Rambo III]]'' (1988), ''[[Rambo (2008 film)|Rambo]]'' (2008), and ''[[Rambo: Last Blood]]'' (2019). Rambo is a [[United States Army Special Forces]] veteran played by [[Sylvester Stallone]], whose [[Vietnam War]] experience traumatized him but also gave him superior military skills, which he has used to fight corrupt [[police officer]]s, enemy troops and [[drug cartel]]s. ''First Blood'' is an adaptation of the 1972 novel ''[[First Blood (novel)|First Blood]]'' by [[David Morrell]].

427 [[Sylvester Stallone]] is the only actor to have appeared in all five films.

428 | colspan=&quot;2&quot; | [[Michael Kozoll]], William Sackheim &amp; [[Sylvester Stallone]]

429 | [[James Cameron]] &amp; Sylvester Stallone

430 | colspan=&quot;2&quot; | [[Sheldon Lettich]] &amp; Sylvester Stallone

431 | Sylvester Stallone

432 | colspan=&quot;2&quot; | Art Monterastelli &amp; Sylvester Stallone

433 | Matthew Cirulnick &amp; Sylvester Stallone

434 | [[Dan Gordon (screenwriter)|Dan Gordon]] &amp; Sylvester Stallone

435 In June 2019 at Cannes, Stallone stated he would continue portraying John Rambo if the fifth film succeeds.&lt;ref name=&quot;Cannes_JB&quot;&gt;{{cite web|url=https://www.joblo.com/horror-movies/news/stallone-would-return-as-rambo-if-last-blood-is-a-hit|title= Stallone happy to return as Rambo if Last Blood is a hit|first=Mike|last=Sprague|date=June 12, 2019|publisher=[[JoBlo.com]]|access-date=June 13, 2019}}&lt;/ref&gt; Grunberg, however, said that ''Rambo: Last Blood'' &quot;closes the circle&quot; and concludes the film series.&lt;ref name=&quot;Total Film&quot;&gt;{{cite web|url=https://www.gamesradar.com/rambo-5-last-blood-director-adrian-grunberg-interview/|title= Exclusive: Rambo: Last Blood director discusses bringing back Sylvester Stallone's action hero: &quot;This movie closes the circle&quot;|author=Jack Shepherd, James Mottram|work=[[GamesRadar]]|date=July 22, 2019|access-date=July 25, 2019}}&lt;/ref&gt; In September 2019, Stallone expressed interest in a sixth ''Rambo'' film, where the character takes refuge at a [[Native American reservation]].&lt;ref name=&quot;Rambo_JB.TV&quot;&gt;{{cite web|title=Sylvester Stallone interview for Rambo: Last Blood!|url=https://www.youtube.com/watch?v=2th50Z9nKcI |archive-url=https://ghostarchive.org/varchive/youtube/20211221/2th50Z9nKcI |archive-date=2021-12-21 |url-status=live|work=[[JoBlo.com]]| date=17 September 2019 |publisher=JoBlo.com official YouTube channel|access-date=18 September 2019|time=3:52}}{{cbignore}}&lt;/ref&gt; Stallone later elaborated that the story would explore Rambo's return to the reservation where he had grown up.&lt;ref name=&quot;Rambo_NTV&quot;&gt;{{Cite web|url=https://www.youtube.com/watch?v=otyZ9qS3-cY |archive-url=https://ghostarchive.org/varchive/youtube/20211221/otyZ9qS3-cY |archive-date=2021-12-21 |url-status=live|title=ランボー ラスト・ブラッド シルベスター・スタローン主演 インタビュー最後の勇者編 Rambo: Last Blood Sylvester Stallone interview movie|date=2020-06-23|work=[[Nippon TV]] Sukkiri|publisher=Shigz Channel, YouTube|access-date=June 25, 2020|time=8:30}}{{cbignore}}&lt;/ref&gt;

436 In August 2013, it was announced that a ''Rambo'' television series was in development. The project was stated to include Avi Lerner and Sylvester Stallone in creative capacities. Stallone was contractually in negotiations to reprise his role from the film series, with John Morayniss and Lerner serving as executive producers. Lerner described the project as &quot;the next phase of the ''Rambo'' legacy.&quot; The series was stated to be a joint-venture production between [[Millennium Media|Nu Image, Millennium Media]] and [[Entertainment One|eOne]], while being shopped around to various [[Television broadcasting|television networks]].&lt;ref name=&quot;RamboTV_Deadline&quot;&gt;{{cite web|url=https://deadline.com/2013/08/entertainment-one-nu-image-to-produce-rambo-tv-series-sylvester-stallone-could-reprise-role-568469/|work=Deadline|title='Rambo' TV Series In Works; Sylvester Stallone In Talks To Reprise Role|author=Andreeva, Nellie|date=August 21, 2013|accessdate=November 6, 2021}}&lt;/ref&gt; By December 2015, the series was ordered by [[Fox Broadcasting Company|FOX Network]] and was officially titled ''Rambo: New Blood''. Written by [[Jeb Stuart (writer)|Jeb Stuart]], the series would center around the complex relationship between John Rambo and an ex-[[United States Navy SEALs|Navy SEAL]] named J.R., his estranged son. Stallone and Stuart joined the production as additional executive producers.&lt;ref name=&quot;RamboTV_Deadline2&quot;&gt;{{cite web|url=https://deadline.com/2015/12/rambo-new-blood-tv-series-sylvester-stallone-fox-father-son-1201635077/|work=Deadline|title='Rambo: New Blood' Father-Son TV Series In Works At Fox With Sylvester Stallone|author=Andreeva, Nellie|date=December 1, 2015|accessdate=November 6, 2021}}&lt;/ref&gt; Later that month, Stallone stated that he would pass on reprising his role.&lt;ref name=&quot;RamboTV_MW&quot;&gt;{{cite web|url=https://movieweb.com/rambo-tv-show-new-blood-no-sylvester-stallone/|work=MovieWeb|title=Rambo TV Show Won't Include Sylvester Stallone|author=Orange, B. Alan|date=December 7, 2015|accessdate=November 6, 2021}}&lt;/ref&gt;

437 In September 2019, Stallone announced plans for a prequel project. Though he would not reprise the title role, he would like to explore who Rambo was before the war: &quot;the perfect guy&quot;, a [[Jim Thorpe]]-type super athlete captain of the sports team, and popular kid at school. Stallone stated that the series would explore the [[juxtaposition]] that took place, to show that joining the military and involved in war, completely changed John Rambo.&lt;ref name=&quot;Rambo_SR&quot;&gt;{{cite web|last=Zinski|first=Dan |title=Sylvester Stallone Wants a Rambo Prequel|url=https://screenrant.com/rambo-prequel-movie-idea-sylvester-stallone/ |website=[[Screen Rant]] |date=16 September 2019 |access-date=17 September 2019}}&lt;/ref&gt; Stallone later teamed with author Chuck Dixon to tell this story in the form of a graphic novel entitled &quot;First Kill&quot;.&lt;ref name=&quot;Rambo.prequel_SR&quot;&gt;{{cite web|url=https://screenrant.com/rambo-movie-future-streaming-prequel-sylvester-stallone/|work=Screen Rant|title=Rambo Can Only Continue As A Streaming Prequel Says Stallone|author=Zinski, Dan|date=November 20, 2020|accessdate=November 6, 2021}}&lt;/ref&gt;

438 In November 2021, Head of Millennium Media Jeffrey Greenstein announced that the studio is once again actively developing a ''Rambo'' television series.&lt;ref name=&quot;RamboTV_THR&quot;&gt;{{cite web|url=https://www.hollywoodreporter.com/movies/movie-news/afm-millennium-media-expendables-4-rambo-tv-1235043020/|work=The Hollywood Reporter|title=AFM: Millennium Boss on Casting 'Expendables 4' and Small-Screen Plans ('Rambo' TV?)|author=Ritman, Alex|date=November 5, 2021|accessdate=November 6, 2021}}&lt;/ref&gt; By November 2022, Stallone stated that development on the prequel series is ongoing. He said that he would like the project to resemble a Vietnam documentary, with the series showing how war changes a person. The plot has John Rambo change from being popular athlete, to what he became in the film series. He also stated that the project may end in modern-day, where he &quot;pass[es] the torch&quot;, which may resemble the original ''New Blood'' project.&lt;ref name=&quot;RamboTV_THR2&quot;&gt;{{cite web|url=https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/|work=The Hollywood Reporter|title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot;|author=Hibberd, James|date=November 7, 2022|accessdate=November 8, 2022}}&lt;/ref&gt;

439 | colspan=&quot;5&quot; | [[Sylvester Stallone]]

440 * ''First Kill'': in November 2022 an Indiegogo campaign launched for a new graphic novel depicting Rambo's first tour of duty in Vietnam written by Sylvester Stallone and [[Chuck Dixon]].

441 The film was scheduled to be released in October 2020, with Shroff expected to star in Hindi remakes of all five films in the ''Rambo'' franchise.&lt;ref&gt;{{cite news |last1=Dixit |first1=Ayush Mohan |title=Sylvester Stallone's Rambo remake starring Tiger Shroff to release on Gandhi Jayanti 2020 {{!}} Bollywood News |url=https://www.timesnownews.com/entertainment/news/bollywood-news/article/sylvester-stallone-s-rambo-remake-starring-tiger-shroff-to-release-on-gandhi-jayanti/420401 |access-date=31 May 2019 |work=[[Times Now]] |date=17 May 2019}}&lt;/ref&gt; Release has been pushed back to 2025 and [[Rohit Dhawan]] is slated to direct the film.&lt;ref&gt;{{cite news |title=Tiger Shroff teams up with Janhvi Kapoor for Rambo remake |url=https://timesofindia.indiatimes.com/entertainment/hindi/bollywood/news/tiger-shroff-teams-up-with-janhvi-kapoor-for-rambo-remake/articleshow/103628123.cms |work=The Times of India |date=13 September 2023}}&lt;/ref&gt;

442 * ''[[Mortal Kombat 11]]'': a [[fighting game]] in which Rambo is a playable DLC character via [[downloadable content]], with Sylvester Stallone reprising his role.&lt;ref&gt;{{cite web |title=Mortal Kombat 11 Ultimate brings Rain, Mileena, and Rambo |url=https://blog.playstation.com/2020/10/08/mortal-kombat-11-ultimate-brings-rain-mileena-and-rambo/ |publisher=PlayStation Blog |access-date=8 October 2020 |date=October 8, 2020}}&lt;/ref&gt; In his arcade ending, Rambo defeated Kronika and planned to use her hourglass to right every wrong in history. As picking who lives and who dies would &quot;kill his soul&quot;, Rambo walked away forever this time, stating &quot;After all I've been through, I've earned a little peace&quot;.

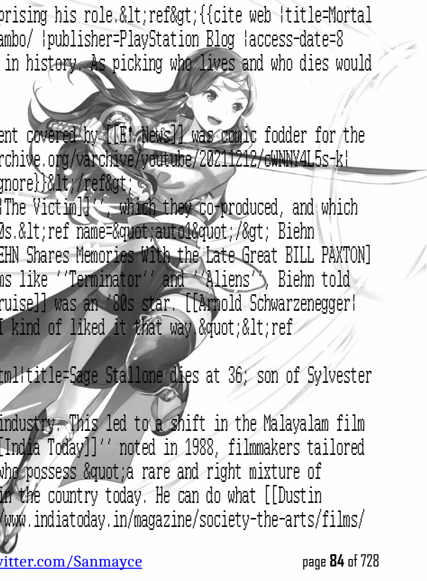
443 {{Sylvester Stallone}}

444 Talk show guests were not the only ones the show skewered. Footage of [[Sylvester Stallone]]'s mother, [[Jackie Stallone]], eating shrimp at an event covered by [[E! News]] was comic fodder for the show.&lt;ref&gt;{{cite web|url=https://www.youtube.com/watch?v=cWNNY4L5s-k&amp;list=PLDA4sT60avlrNurJelpnC3CNRXB2s3Tz- |archive-url=https://ghostarchive.org/varchive/youtube/20211212/cWNNY4L5s-k|archive-date=2021-12-12 |url-status=live|title=Jackie Stallone Eats Jumbo Shrimp |publisher=YouTube |date=2007-08-12 |access-date=2013-09-03}}{{cbignore}}&lt;/ref&gt;

445 [[File:8.23.12BiehnBlancByLuigiNovi3.jpg|thumb|Biehn and his future wife, actress [[Jennifer Blanc]], in 2012, promoting ''[[The Victim (2011 film)|The Victim]]'', which they co-produced, and which Biehn starred in and directed]] Coinciding with the decline of his acting career during the 1990s, Biehn suffered from alcoholism until the mid-2000s.&lt;ref name=&quot;auto1&quot;/&gt; Biehn suffered a [[stroke]] around 2008 and had [[Cardiac surgery|open heart surgery]].&lt;ref&gt;[https://www.youtube.com/watch?v=wnPyt4Mo8Do MICHAEL BIEHN Shares Memories With the Late Great BILL PAXTON] Inside Of You Clips, April 4, 2023.&lt;/ref&gt; When questioned on why he did not attain stardom despite having lead roles in well-regarded hit films like ''Terminator'' and ''Aliens'', Biehn told the ''Hollywood Reporter'' &quot;People always talk about me being an '80s star. I was not an '80s star. [[Bruce Willis]] was an '80s star. [[Tom Cruise]] was an '80s star. [[Arnold Schwarzenegger|Schwarzenegger]] and [[Sylvester Stallone|Stallone]]. [[Mel Gibson]]. Those guys were making $20 million [a picture]. I never even got $1 million. I kind of liked it that way.&quot;&lt;ref name=&quot;auto&quot;/&gt;

446 * [[Sage Stallone]], actor and producer&lt;ref&gt;{{cite web|url=https://www.latimes.com/local/obituaries/la-me-0714-sage-stallone-20120714-story.html|title=Sage Stallone dies at 36; son of Sylvester Stallone|first=Bob |last=Pool |date=2012-07-14|website=[[Los Angeles Times]]|access-date=14 March 2018}}&lt;/ref&gt;

447 During the 1980s, Mohanlal and [[Mammootty]] emerged as &quot;superstars&quot; in [[Malayalam cinema]], marking an unprecedented phenomenon in the industry. This led to a shift in the Malayalam film landscape, with [[Star vehicle|star-driven]] films becoming the norm, reminiscent of [[Bollywood]] and [[Tamil cinema|Tamil]] film cultures. As ''[[India Today]]'' noted in 1988, filmmakers tailored characters to suit these superstars. Director [[I. V. Sasi]] remarked that even [[Prem Nazir]] never had the popularity of Mohanlal and Mammootty, who possess &quot;a rare and right mixture of charisma and acting ability&quot;. In 1988, filmmaker [[Fazil (director)|Fazil]] stated that, &quot;Mohanlal is the most flexible and subtle actor in the country today. He can do what [[Dustin Hoffman]] did in ''[[Tootsie]]'' or [[Silvester Stallone]] did in ''[[Rambo (franchise)|Rambo]]'' with ease&quot;.&lt;ref&gt;{{Cite web|url=https://www.indiatoday.in/magazine/society-the-arts/films/



story/19881231-mammootty-and-mohan-lal-emerge-as-unprecedented-superstars-of-malayalam-films-798061-1988-12-31|archive-url = https://web.archive.org/web/20190918142917/https://www.indiatoday.in/magazine/society-the-arts/films/story/19881231-mammootty-and-mohan-lal-emerge-as-unprecedented-superstars-of-malayalam-films-798061-1988-12-31|archive-date = 18 September 2019|title = Mammootty and Mohan Lal emerge as unprecedented superstars of Malayalam films| date=31 December 1988 }}&lt;/ref&gt;

448 By 1982, promoter [[Don King (boxing promoter)|Don King]] and manager Dennis Rappaport began one of the most massive and racially toned campaigns in boxing history to raise public interest for a fight between Holmes and Cooney. After they were both signed to fight, an intense promotional tour followed.{{citation needed|date=June 2020}} Holmes and Cooney attended press conferences at several [[United States]] cities, Cooney was shown on the cover of ''[[Time magazine|Time]]'' magazine, [[Cinema of the United States|Hollywood]] stars took an interest in the fight ([[Sylvester Stallone]] in particular hung out with Gerry Cooney, others, such as [[Woody Allen]], attended the fight live) and Cooney was cast as &quot;The Great White Hope&quot;.&lt;ref&gt;{{cite news|url=https://www.independent.co.uk/sport/general/boxing/gerry-cooney-larry-holmes-muhammad-ali-george-foreman-heavyweights-a8972211.html|title=To the brink and back: The story behind Gerry Cooney's road to recovery|date=24 June 2019|work=[[The Independent]]|accessdate=2020-06-23}}&lt;/ref&gt; There had not been a White world Heavyweight champion in 22 years and Cooney would try to change that.{{citation needed|date=June 2020}} White supremacist groups had announced they would have &quot;agents&quot; ready to shoot at Holmes the moment he entered the ring, and Black groups retaliated by answering that they would also have armed people on hand in case Holmes was attacked.{{citation needed|date=June 2020}} Because of this, there were [[Sniper#Law enforcement applications|police snipers]] on the roofs of every major hotel surrounding the venue, once again, the Caesars Palace hotel and casino.&lt;ref name=&quot;auto&quot;&gt;{{Cite web|url=https://www.ringtv.com/640367-larry-holmes-gerry-cooney-fanning-the-flames/|title=Larry Holmes-Gerry Cooney: Fanning the Flames|first=The|last=Ring|date=June 11, 2022}}&lt;/ref&gt; Snipers were used because the fight was held in the hotel's parking lot; any attacker could have been easily shot by police snipers.&lt;ref name=&quot;auto&quot; /&gt;

449 In 1996, Harris shared dual roles with [[Katherine Heigl]] for the television film ''[[Wish Upon a Star]]'': Harris played Hayley Wheaton, a nerdy girl who [[body swap|switches bodies]] with her older, more popular sister Alexia (Heigl). Also in 1996, Harris starred in the films ''Shattered Image'' and ''[[Back to Back (1996 film)|Back to Back]]'', and, the same year, had the role of young survivor Ashley Crighton in ''[[Daylight (1996 film)|Daylight]]'', the disaster film toplined by [[Sylvester Stallone]]. While ''Daylight'' has a 26% approval rating on Rotten Tomatoes&lt;ref&gt;{{cite web|url=https://www.rottentomatoes.com/m/daylight/ |title=Daylight |website=Rotten Tomatoes |date=December 6, 1996 |access-date=September 25, 2012}}&lt;/ref&gt; and grossed $33 million in the United States, it took in over $126 million overseas, resulting in gross earnings of $159,212,469 worldwide.&lt;ref&gt;{{cite web|url=https://www.boxofficemojo.com/movies/?id=daylight.htm |title=Daylight (1996) |website=Box Office Mojo |date=January 17, 1997 |access-date=September 25, 2012}}&lt;/ref&gt; In 1997, Harris appeared in two episodes of the medical drama ''[[ER (TV series)|ER]]'' as Laura Quentin. In 1998, Harris had the lead role of Lulu in the film ''Dizzyland'', where she portrayed a sexually abused teenager, and also appeared in an episode of ''[[Diagnosis: Murder]]''. She then appeared in the popular slasher ''[[Urban Legend (film)|Urban Legend]]'', her first horror film since her early ''Halloween'' credits. She portrayed Tosh, a [[goth subculture|goth girl]] who is murdered while her roommate Natalie ([[Alicia Witt]]) is resting on the other side of the room.{{citation needed|date=July 2021}}

450 [[File:Godkiller Danielle-Harris chase variant cover directed by Matt Pizzolo.jpg|thumb|left|185px|Harris in 2010]]{{Quote box |width=330px |bgcolor=LightCyan |align=right | quote=My career has not been easy for me. I feel like after everything I've done, I've kind of had to start over. I remember a couple of months ago... I found a ''[[People (magazine)#Teen People|Teen People]]'' magazine with me on the cover of &quot;The New It Girls&quot;, and it was me, [[Jessica Biel]], [[Scarlett Johansson]] and [[Kirsten Dunst]]. I was in that company at one point in my career - I had just done ''[[Daylight (1996 film)|Daylight]]'' with [[Sylvester Stallone]] [in 1996]. And then somewhere along the line, I don't know what happened. It was not for a lack of trying - they got that one movie after that that really pushed them, and I feel like I'm always looking for that movie. I've just managed to barely hang in there by the skin of my teeth, but I think that's why I'm still working is because I've hung in there. With [[Rob Zombie]] bringing me back for ''[[Halloween (2007 film)|Halloween]]'', it's ironic that the same movie that started my career reinvented it again.|source=--Harris on becoming a &quot;scream queen&quot;&lt;ref&gt;{{cite web|url=http://vip.usaweekend.com/2010/10/hatchet-ii-star-danielle-harris-adores-her-role-as-cinematic-scream-queen/ |archive-url=https://archive.today/20130412025153/http://vip.usaweekend.com/2010/10/hatchet-ii-star-danielle-harris-adores-her-role-as-cinematic-scream-queen/ |title='Hatchet II' star Danielle Harris adores her role as cinematic 'scream queen' |website=USA Weekend |date=1 October 2010 |archive-date=April 12, 2013 |access-date=21 September 2015}}&lt;/ref&gt;}}

451 * [[Sylvester Stallone|Sylvester]] and [[Frank Stallone]]; actors and musician, respectively

452 Other figures pied by the group included [[Jean Charest]],&lt;ref name=charestdumont&gt;[http://ici.radio-canada.ca/nouvelles/Index/nouvelles/200304/12/003-CHARESTENTARTE.shtml &quot;Mario Dumont et Jean Charest entartés&quot;]. [[CBC News|Radio-Canada]], April 13, 2003.&lt;/ref&gt; [[Mario Dumont]],&lt;ref name=charestdumont /&gt; [[Stéphane Dion]],&lt;ref name=mallick&gt;[https://www.theglobeandmail.com/arts/cream-and-punishment/article4162784/?page=all &quot;Cream and punishment&quot;]. ''[[The Globe and Mail]]'', April 18, 2000.&lt;/ref&gt; [[Allan Rock]],&lt;ref name=chretien&gt;[https://www.theglobeandmail.com/news/national/chretien-gets-pie-in-face/article4166870/ &quot;Chrétien gets pie in face&quot;]. ''[[The Globe and Mail]]'', August 17, 2000.&lt;/ref&gt; [[Pierre Pettigrew]],&lt;ref name=snipers /&gt; [[William Johnson (author)|William Johnson]],&lt;ref name=snipers /&gt; [[Pierre Bourque (politician)|Pierre Bourque]],&lt;ref name=snipers /&gt; [[Jean Doré]],&lt;ref&gt;&quot;Doré entarté en pleine Fin du monde&quot;. ''[[La Presse (Canadian newspaper)|La Presse]]'', October 29, 1998.&lt;/ref&gt; [[Jacques Parizeau]],&lt;ref name=mallick/&gt; [[Bernard Landry]]&lt;ref name=mallick /&gt; and [[Jacques Duchesneau]].&lt;ref name=snipers /&gt; Quebec politicians were the most common targets, although the group also pied [[Sylvester Stallone]] during a 1998 promotional tour.&lt;ref name=snipers /&gt;

453 [[File:Sylvester Stallone Cannes 2019.jpg|thumb|right|150px|[[Sylvester Stallone]], Worst Supporting Actor winner]]

454 | style=&quot;background:#B0C4DE;&quot; |'''[[Sylvester Stallone]] in ''[[Spy Kids 3-D: Game Over]]'' as The Toymaker'''

455 ******** Nielsen has been married five times. She was first married to composer [[Kasper Winding]] (see [[#W|Winding]]). Their son is television personality, [[Julian Winding]]. She has also been married to actor [[Sylvester Stallone]] (see [[#S|Stallone]]), television writer [[Raoul Meyer]] and producer [[Mattia Dessi]].

456 * Actor, writer, and hairstylist [[Frank Stallone Sr.]] is the father of actors [[Sylvester Stallone]] and [[Frank Stallone]] with first wife and astrologer [[Jackie Stallone]].

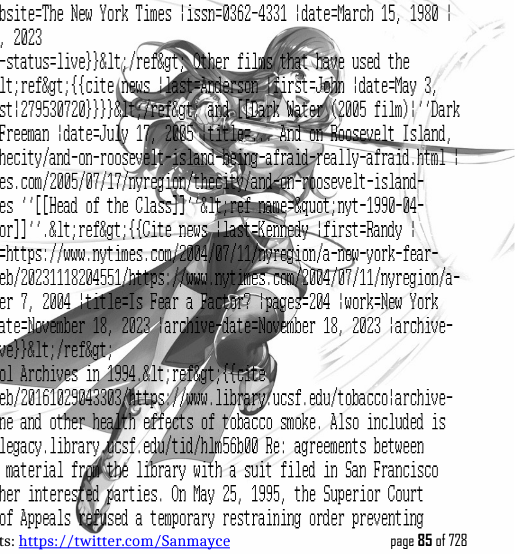
457 ***** Nielsen was married to actor [[Sylvester Stallone]] (see [[#S|Stallone]]), director and photographer [[Sebastian Copeland]] (see [[#C|Casadesus]]), television writer [[Raoul Meyer]] and producer [[Mattia Dessi]].

458 !scope=&quot;row&quot;|[[Marvelous Marvin Hagler]]&lt;br /&gt;[[Wayne Newton]]&lt;br /&gt;[[Sylvester Stallone]]&lt;br /&gt;[[Arnold Schwarzenegger]]&lt;br /&gt;[[Bruce Willis]]

459 ''The New York Times'' wrote in 2006 that the tramway was not only a vital mode of transport for Roosevelt Island residents, particularly students and the elderly, but also &quot;offers a rare chance to live purely in the moment in New York, no small achievement&quot;.&lt;ref&gt;{{Cite news |last=Dunning |first=Jennifer |date=April 30, 2006 |title=Riders in the Sky |language=en-US |work=The New York Times |issn=0362-4331 |url=https://www.nytimes.com/2006/04/30/nyregion/thecity/riders-in-the-sky.html |access-date=November 18, 2023 |archive-date=November 18, 2023 |archive-url=https://web.archive.org/web/20231118222605/https://www.nytimes.com/2006/04/30/nyregion/thecity/riders-in-the-sky.html |url-status=live}}&lt;/ref&gt; The ''Times'' attributed the tram's popularity to its relative safety and comfort, its lack of overcrowding, and the views from the cabins.&lt;ref name=&quot;nyt20080115&quot; /&gt; Over the years, the tram attracted notable riders such as actor [[Harrison Ford]], tennis player [[Andre Agassi]], and actor/filmmaker [[Sylvester Stallone]].&lt;ref name=&quot;Parco 2016 r111&quot; /&gt; After the tramway was renovated, a writer for ''[[The Washington Post]]'' said that the line allowed a &quot;pretty awesome&quot; view of the skyline.&lt;ref name=&quot;Yarvin 2011&quot; /&gt; A ''[[Curbed]]'' writer said in 2017 that, though the tramway's usefulness was limited by the fact that it traveled only between Manhattan and Roosevelt Island, this was counterbalanced by its &quot;sheer awesomeness&quot; and the fact that it was less crowded than the subway.&lt;ref&gt;{{cite web |last=Plitt |first=Amy |date=September 20, 2017 |title=Roosevelt Island Tram is New York's coolest way to get around |url=https://ny.curbed.com/2017/9/20/16271924/nyc-roosevelt-island-tram-coolest-transit |access-date=November 23, 2023 |website=Curbed NY |archive-date=March 31, 2023 |archive-url=https://web.archive.org/web/20230331152331/https://ny.curbed.com/2017/9/20/16271924/nyc-roosevelt-island-tram-coolest-transit |url-status=live}}&lt;/ref&gt;

460 The tramway has also been used as a filming location. The 1981 film [[Nighthawks (1981 film)|''Nighthawks'']] was partially set on the tramway;&lt;ref&gt;{{cite web |last=Maslin |first=Janet |date=April 10, 1981 |title='Nighthawks' With Sylvester Stallone |url=https://www.nytimes.com/1981/04/10/movies/nighthawks-with-sylvester-stallone.html |access-date=November 21, 2023 |website=The New York Times |issn=0362-4331 |archive-date=June 25, 2023 |archive-url=https://web.archive.org/web/20230625051648/https://www.nytimes.com/1981/04/10/movies/nighthawks-with-sylvester-stallone.html |url-status=live}}&lt;/ref&gt; many residents objected when the filming of one scene required shutting down the tram for a few days.&lt;ref&gt;{{cite news |id={{ProQuest|162705130}} |title=Stallone New Film Angers Roosevelt Island Residents |last=Taylor |first=Clarke |date=March 18, 1980 |page=G6 |work=Los Angeles Times |issn=0458-3035|postscript=none}}; {{Cite news |last=Edmonds |first=Richard |date=March 18, 1980 |title=Stallone Flick Grounds Roosevelt Tram Riders |pages=4 |work=Daily News |via=newspapers.com |url=https://www.newspapers.com/article/daily-news-stallone-flick-grounds-roosev/135521233/ |access-date=November 21, 2023 |archive-date=November 21, 2023 |archive-url=https://web.archive.org/web/20231121020528/https://www.newspapers.com/article/daily-news-stallone-flick-grounds-roosev/135521233/ |url-status=live|postscript=none}}; {{Cite web |last=Smolowe |first=Jill |title=Movie to Shut Tram 3 Days |website=The New York Times |issn=0362-4331 |date=March 15, 1980 |url=https://www.nytimes.com/1980/03/15/archives/movie-to-shut-tram-3-days.html |access-date=November 21, 2023 |archive-date=November 21, 2023 |archive-url=https://web.archive.org/web/20231121020526/https://www.nytimes.com/1980/03/15/archives/movie-to-shut-tram-3-days.html |url-status=live}}&lt;/ref&gt; Other films that have used the tramway as a setting include Sam Raimi's ''[[Spider-Man (2002 film)|Spider-Man]]'' (2002)&lt;ref name=&quot;nyt-2003-04-27&quot; /&gt;&lt;ref&gt;{{cite news |last=Anderson |first=John |date=May 3, 2002 |title='Spider-Man' Is Just Marvel-ous / A fun spin on action, humor, romance |page=B11 |work=Newsday |issn=2574-5298 |id={{ProQuest|279530720}}}}&lt;/ref&gt; and [[Dark Water (2005 film)|''Dark Water'']] (2005), directed by Walter Salles, the latter of which was filmed on the route.&lt;ref&gt;{{Cite news |last=Gill |first=John Freeman |date=July 17, 2005 |title=... And on Roosevelt Island, Being Afraid, Really Afraid |language=en-US |work=The New York Times |issn=0362-4331 |url=https://www.nytimes.com/2005/07/17/nyregion/thecity/and-on-roosevelt-island-being-afraid-really-afraid.html |access-date=November 18, 2023 |archive-date=November 18, 2023 |archive-url=https://web.archive.org/web/20231118222603/https://www.nytimes.com/2005/07/17/nyregion/thecity/and-on-roosevelt-island-being-afraid-really-afraid.html |url-status=live}}&lt;/ref&gt; In addition, the tramway was shown in the opening credits of the TV series ''[[Head of the Class]]''&lt;ref name=&quot;nyt-1990-04-08&quot; /&gt; and ''[[Archie Bunker's Place]]'',&lt;ref name=&quot;Newsday 1986&quot; /&gt; as well as a 2004 episode of ''[[Fear Factor]]''.&lt;ref&gt;{{Cite news |last=Kennedy |first=Randy |date=July 11, 2004 |title=A New York 'Fear Factor'? Try That Last Hot Dog |language=en-US |work=The New York Times |issn=0362-4331 |url=https://www.nytimes.com/2004/07/11/nyregion/a-new-york-fear-factor-try-that-last-hot-dog.html |access-date=November 18, 2023 |archive-date=November 18, 2023 |archive-url=https://web.archive.org/web/20231118204551/https://www.nytimes.com/2004/07/11/nyregion/a-new-york-fear-factor-try-that-last-hot-dog.html |url-status=live|postscript=none}}; {{Cite news |last=Hatty |first=Michele |date=November 7, 2004 |title=Is Fear a Factor? |pages=204 |work=New York Daily News |issn=2692-1251 |via=newspapers.com |url=https://www.newspapers.com/article/daily-news-is-fear-a-factor/135396915/ |access-date=November 18, 2023 |archive-date=November 18, 2023 |archive-url=https://web.archive.org/web/20231118204551/https://www.newspapers.com/article/daily-news-is-fear-a-factor/135396915/ |url-status=live}}&lt;/ref&gt;

461 Thousands of pages of B&amp;W documents were donated unsolicited to the [[University of California San Francisco]] (UCSF) Tobacco Control Archives in 1994.&lt;ref&gt;{{cite web|url=http://www.library.ucsf.edu/tobacco|title=Tobacco Control Archives|access-date=2010-05-30|archive-url=https://web.archive.org/web/20161029043303/https://www.library.ucsf.edu/tobacco|archive-date=2016-10-29|url-status=dead}}&lt;/ref&gt; These documents consist primarily of scientific studies on the addictive nature of nicotine and other health effects of tobacco smoke. Also included is documentation of $500,000 in payments to [[Sylvester Stallone]] for promoting B&amp;W products in five of his films.&lt;ref&gt;[http://legacy.library.ucsf.edu/tid/hlm56b00 Re: agreements between Stallone and Associated Film Promotions] Legacy Tobacco Documents Library&lt;/ref&gt; B&amp;W sought to permanently remove the disputed material from the library with a suit filed in San Francisco Superior Court. The university contended that all of the documents were in the public domain and should be available to scholars and other interested parties. On May 25, 1995, the Superior Court ruled that these documents should be made available for public review. B&amp;W appealed that decision, and on June 23, 1995, the Court of Appeals refused a temporary restraining order preventing



release of the documents. On June 29, the California Supreme Court rejected the company's appeal, allowing UCSF to release the documents.&lt;ref&gt;{{cite web¦url=https://www.industrydocumentslibrary.ucsf.edu/tobacco/¦title=Industry Documents Library¦website=www.industrydocumentslibrary.ucsf.edu}}&lt;/ref&gt;

462 At age five, Clay was entertaining his family with [[impressionist (entertainment)¦impression]]s, and by seven he was playing drums, inspired by the sounds of [[big band music]].&lt;ref&gt;{{cite news¦url=https://www.newspapers.com/image/186215526/¦title=Dice takes his chops¦date=3 June 1988¦newspaper=Philadelphia Daily News¦page=42¦via=[[Newspapers.com]]¦url-access=subscription ¦access-date=14 April 2018}}&lt;/ref&gt;&lt;ref name=latimes1987/&gt; Clay attended [[James Madison High School (Brooklyn)¦James Madison High School]] in Brooklyn and as a teenager worked as a drummer on the [[Catskill Mountains]] circuit in the 1970s, playing [[bar mitzvah]]s and weddings under the name Clay Silvers.&lt;ref name=latimes1987/&gt;&lt;ref name=latimes1990/&gt;&lt;ref name=desmoines1992&gt;{{cite news¦url=https://www.newspapers.com/image/132063947/¦title=A Dicey situation¦first=Deborah¦last=Cushman¦date=5 November 1992¦newspaper=The Des Moines Register¦page=59¦via=[[Newspapers.com]]¦url-access=subscription ¦access-date=14 April 2018}}&lt;/ref&gt; Upon returning to New York City, he failed to form a big band of his own. He pursued higher education but dropped out to become a full-time comedian.&lt;ref name=latimes1987/&gt;&lt;ref name=desmoines1992/&gt; Clay cites [[Elvis Presley]], [[Fonzie]], [[John Travolta]], and [[Sylvester Stallone]] as his heroes.&lt;ref name=latimes1990/&gt;

463 * [[Sylvester Stallone]] (February 27, 2005)

464 By the early 1980s, Winner found himself in great need of a successful film and accepted [[Charles Bronson]]'s request to film ''[[Death Wish II]]'' (1981), a sequel to the 1974 hit. Bronson had already signed a lucrative deal with [[Cannon Films]], independent producer of exploitation fare and marginal art house titles. The sequel, co-starring Bronson's wife [[Jill Ireland]], considerably increased the violence to more graphic levels. Winner said the film was 'the same, but different', to the original. 'That's what sequels are - ''[[Rocky II]]'', ''[[Rocky III]]'' - you don't see [[Sylvester Stallone]] move to the [[Congo Basin¦Congo]] and become a nurse. Here the look of LA is what's different. Besides - rape doesn't date!'&lt;ref name=&quot;death&quot;&gt;{{cite news¦title=THE REINCARNATION OF A 'DEATH WISH'¦author=Trombetta, Jim¦work=Los Angeles Times¦date=13 July 1981¦page=g1}}&lt;/ref&gt; ''Death Wish II'' made a $2 million profit for Cannon films&lt;ref name=&quot;yule&quot;&gt;Andrew Yule, ''Hollywood a Go-Go: The True Story of the Cannon Film Empire'', Sphere Books, 1987 p24&lt;/ref&gt; and made an extra $29 million worldwide.

465 Between the two ''Star Wars'' films, he starred alongside [[Sylvester Stallone]] as a cop in the thriller ''[[Nighthawks (1981 film)¦Nighthawks]]'' (1981). Williams returned to Broadway in the [[August Wilson]] play ''[[Fences (play)¦Fences]]'', as a replacement for [[James Earl Jones]] in the role of Troy Maxson in 1988.&lt;ref&gt;{{Citation ¦ last = Falkner¦ first = David ¦ title = The Actor as Athlete: Subtle and Complex Portrait ¦ newspaper = [[The New York Times]] ¦ date = February 7, 1988 ¦ url = https://www.nytimes.com/1988/02/07/theater/the-actor-as-athelte-subtle-and-complex-portrait.html?pagewanted=all ¦ access-date = January 3, 2009}}&lt;/ref&gt; Williams co-starred in [[Batman (1989 film)¦1989's ''Batman'']] as district attorney [[Harvey Dent (1989 film series character)¦Harvey Dent]], a role that was planned to develop into Dent's alter-ego, the villain Two-Face, in sequels. He was set to reprise the role in the sequel ''[[Batman Returns]]'', but his character was deleted and replaced with villain Max Shreck. When [[Joel Schumacher]] stepped in to direct ''[[Batman Forever]]'', where Two-Face was to be a secondary villain, Schumacher decided to hire [[Tommy Lee Jones]] for the role.&lt;ref&gt;{{cite web ¦first=David ¦last=Bentley ¦url=http://blogs.coventrytelegraph.net/thegeekfiles/2008/07/aaron-eckhart-on-creating-the.html ¦title=Aaron Eckhart on creating the new face of Two Face ¦work=Coventry Telegraph ¦date=July 2008 ¦archive-url=https://web.archive.org/web/20130524180640/http://blogs.coventrytelegraph.net/thegeekfiles/2008/07/aaron-eckhart-on-creating-the.html ¦archive-date=May 24, 2013 }}&lt;/ref&gt; There was a rumor that Schumacher had to pay Williams a fee in order to hire Jones, but Williams said that it was not true: &quot;You only get paid if you do the movie. I had a two-picture deal with ''Star Wars.'' They paid me for that, but I only had a one picture deal for ''Batman.''&quot;&lt;ref&gt;{{cite web¦url=http://comicbook.com/blog/2013/11/02/billy-dee-williams-talks-two-face-did-not-get-paid-for-batman-forever/¦title=Billy Dee Williams Talks Two-Face, Did Not Get Paid For Batman Forever¦publisher=Comicbook.com¦date=November 2, 2013¦access-date=September 20, 2014}}&lt;/ref&gt; Williams eventually voiced Two-Face in the 2017 film ''[[The Lego Batman Movie]]''.&lt;ref name=&quot;Billy&quot;&gt;{{Cite news¦first=Patrick¦last=Shanley¦url=http://www.hollywoodreporter.com/heat-vision/billy-dee-williams-star-wars-lunch-donald-glover-lando-calrissian-971886¦title=Billy Dee Williams on Secret 'Star Wars' Lunch with Donald Glover and Finally Playing Two-Face¦date=February 2, 2017¦newspaper=Hollywood Reporter¦language=en-US¦access-date=February 4, 2017}}&lt;/ref&gt;

466 File:Sylvester Stallone Cannes 2019.jpg¦[[Sylvester Stallone]], Worst Actor of the Century winner.

467 ¦ style=&quot;background:#B0C4DE;&quot; ¦'''[[Sylvester Stallone]] for &quot;99.5% of everything he's ''EVER'' done'''

468 In 1968, she married Jeff Wald, a native of [[the Bronx]].&lt;ref name=NYTobit/&gt; She converted to Judaism before marrying Wald,&lt;ref name=&quot;ref23&quot;&gt;{{cite news ¦last=Levins ¦ first=Harry ¦title=People in the News ¦newspaper=[[St. Louis Post-Dispatch]] ¦date=14 December 2000 ¦url=http://nl.newsbank.com/nl-search/we/Archives?p_product=SL&amp;p_theme=sl&amp;p_action=search&amp;p_maxdocs=200&amp;p_topdoc=1&amp;p_text_direct-0=0EB052D53A6AA236&amp;p_field_direct-0=document_id&amp;p_perpage=10&amp;p_sort=YMD_date:D&amp;s_trackval=GooglePM ¦access-date=16 June 2008}} - &quot;Although Helen Reddy is Jewish, she has just released an album titled 'The Best Christmas Ever.' When an Internet interviewer cocked an eyebrow, Reddy said she had stuck to her religious beliefs by making sure that no song mentioned Jesus.&quot;&lt;/ref&gt; with whom she had a son, Jordan,&lt;ref&gt;{{Cite web¦last=White¦first=Abbey¦date=2021-11-13¦title=Jeff Wald, Producer and Manager for Helen Reddy and Sylvester Stallone, Dies at 77¦url=https://www.hollywoodreporter.com/news/general-news/jeff-wald-dead-producer-manager-boxing-promoter-1235047560/¦access-date=2021-11-14¦website=The Hollywood Reporter¦language=en-US}}&lt;/ref&gt;{{efn¦Original name Jordan Wald, 1972, currently Jordan Sommers}} born in 1972.&lt;ref name=&quot;people1983&quot;&gt;{{cite news¦url=http://www.people.com/people/archive/article/0,,20084995,00.html ¦title=Tug-of-war for a 10&amp;nbsp;year-old ¦work=[[People (magazine)¦People]] ¦first=Jeff ¦last=Jarvis ¦date=16 May 1983 ¦ access-date=20 March 2015}}&lt;/ref&gt;

469 Reddy's second husband and former manager, Jeff Wald, died on November 12, 2021, at the age of 77.&lt;ref&gt;{{cite web ¦ url=https://variety.com/2021/music/news/jeff-wald-dead-helen-reddy-1235111642/ ¦ title=Jeff Wald, Who Managed Helen Reddy, Sylvester Stallone and Donna Summer, Dies at 77 ¦ date=13 November 2021 }}&lt;/ref&gt;

470 In the early 1990s, Rourke was offered and declined the role of [[Butch Coolidge]], which later became [[Bruce Willis]]'s role in ''[[Pulp Fiction (film)¦Pulp Fiction]]''.&lt;ref&gt;{{cite web¦url=https://www.yahoo.com/entertainment/oscars/blog/19-rourke-is-back-but-for-how-long ¦title=Rourke Is Back, But For How Long? ¦publisher=Yahoo! Movies ¦access-date=July 5, 2009 ¦url-status=dead ¦ archive-url=https://web.archive.org/web/20090206005354/http://oscars.movies.yahoo.com/blog/19-rourke-is-back-but-for-how-long?nc ¦archive-date=February 6, 2009 }}&lt;/ref&gt; After his retirement from boxing, Rourke did accept supporting roles in several 1990s films, including [[Francis Ford Coppola]]'s [[The Rainmaker (1997 film)¦adaptation]] of [[John Grisham]]'s ''[[The Rainmaker (John Grisham)¦The Rainmaker]]'', [[Vincent Gallo]]'s ''[[Buffalo '66]]'', [[Steve Buscemi]]'s ''[[Animal Factory]]'', [[Sean Penn]]'s ''[[The Pledge (film)¦The Pledge]]'', and [[Sylvester Stallone]]'s remake of ''[[Get Carter (2000 film)¦Get Carter]]''. Rourke also has written several films under the name Sir Eddie Cook, including ''[[Bullet (1996 film)¦Bullet]]'', in which he co-starred with [[Tupac Shakur]].

471 In 2010, Rourke played the role of the main villain [[Whiplash (comics)#Anton Vanko¦Whiplash]] in the film ''[[Iron Man 2]]''. In an interview with New Zealand magazine ''[[Rip It Up (New Zealand)¦Rip It Up]]'' magazine he revealed that he prepared for the role by visiting Russian jail inmates.&lt;ref&gt;{{cite web ¦url=http://ripitup.co.nz/contentitem/interview-mickey-rourke-talks-iron-man-2/911 ¦title=INTERVIEW: Mickey Rourke Talks Iron Man 2 ¦website=[[Rip It Up (New Zealand)¦Rip It Up]] ¦access-date=15 January 2022 ¦archive-url=https://web.archive.org/web/20120308113645/http://ripitup.co.nz/contentitem/interview-mickey-rourke-talks-iron-man-2/911 ¦archive-date=8 March 2012 ¦url-status=dead}}&lt;/ref&gt; In 2011, he portrayed the villainous King Hyperion in ''[[Immortals (2011 film)¦Immortals]]'' and received praise for his performance, while the film received mixed-to-positive reviews and became a box office success. He also had a minor role as Tool in [[Sylvester Stallone]]'s ''[[The Expendables (2010 film)¦The Expendables]]''. Though he had little screen time, his performance was met with rave reviews and cited as one of the film's highlights.

472 ¦''[[Cliffhanger (film)¦Cliffhanger]]'', screenplay by [[Michael France]] and [[Sylvester Stallone]], screen story by France, based on a premise by [[John Long (climber)¦John Long]]

473 ¦[[Sylvester Stallone]] in ''Driven'' as Joe Tanto

474 ¦[[Burt Reynolds]] and [[Sylvester Stallone]] in ''[[Driven (2001 film)¦Driven]]''

475 ¦''[[Driven (2001 film)¦Driven]]'', screenplay by [[Sylvester Stallone]], story by Jan Skrentny and Neal Tabachnick

476 File:Sylvester Stallone 2012.jpg¦[[Sylvester Stallone]], Worst Actor winner.

477 ¦ style=&quot;background:#B0C4DE;&quot; ¦'''[[Sylvester Stallone]]''' in '''''[[Stop! Or My Mom Will Shoot]]'' as Sgt. Joe Bomowski'''

478 ¦[[Sylvester Stallone]] in ''[[Oscar (1991 film)¦Oscar]]'' as Angelo &quot;Snaps&quot; Provolone

479 ¦[[Sylvester Stallone]] in ''[[Rocky V]]'' as [[Rocky Balboa]]

480 ¦''[[Rocky V]]'', written by [[Sylvester Stallone]]

481 The film was [[greenlight]]ed for production on January 21, 1999.&lt;ref&gt;{{cite news¦url=https://www.variety.com/article/VR1117490459.html?categoryid=3&amp;cs=1¦title=Hoblit time-trips; old script scores for Iliff¦work=[[Variety (magazine)¦Variety]]¦access-date=June 26, 2007¦date=January 21, 1999}}&lt;/ref&gt; [[Sylvester Stallone]] was rumored to be taking the role of Frank Sullivan in 1997, but fell out of the deal after a dispute over his fee.&lt;ref name=&quot;varietyjune1997&quot;&gt;{{cite news ¦last=Cox ¦first=Dan ¦date=June 6, 1997 ¦title=Sly eyeing New Line's 'Frequency' ¦ work=[[Variety (magazine)¦Variety]] ¦url=https://variety.com/1997/film/news/sly-eyeing-new-line-s-frequency-1116679561/ ¦access-date=July 31, 2023 ¦archive-url=https://archive.today/20130412023755/http://variety.com/1997/film/news/sly-eyeing-new-line-s-frequency-1116679561/ ¦archive-date=April 12, 2013}}&lt;/ref&gt;&lt;ref name=&quot;varietyjuly1997&quot;&gt;{{cite news¦url=https://www.variety.com/vstory/VR1117342811.html?categoryid=38&amp;cs=1¦title=INSIDE MOVES¦work=[[Variety (magazine)¦Variety]]¦access-date=July 13, 2007¦last=Busch¦first=Anita M.¦date=June 27, 1997}}&lt;/ref&gt; [[Renny Harlin]] was rumored to be director on the film.&lt;ref name=&quot;varietyjune1997&quot; /&gt;&lt;ref name=&quot;varietyjuly1997&quot; /&gt; [[Gregory Hoblit]] first read the script in November 1997, eighteen months after his father's death. In a 2000 interview shortly after the American release of ''Frequency'', he described the film as &quot;high risk&quot; since the project had already been passed among several directors, including one of note who had twice the budget Hoblit was given.&lt;ref name=&quot;salonmay2000&quot;&gt;{{cite web¦url=http://archive.salon.com/ent/col/srag/2000/05/25/hoblit/index2.html¦title=What's the &quot;Frequency,&quot; Gregory?¦work=[[Salon.com]]¦access-date=June 27, 2007¦last=Sragow¦first=Michael¦date=May 25, 2000}}&lt;/ref&gt; In the same interview, he described the difficulty he had finding the two leads. Hoblit realized he needed an &quot;experienced actor&quot; to portray Frank Sullivan and thus chose Dennis Quaid.&lt;ref name=&quot;salonmay2000&quot; /&gt;

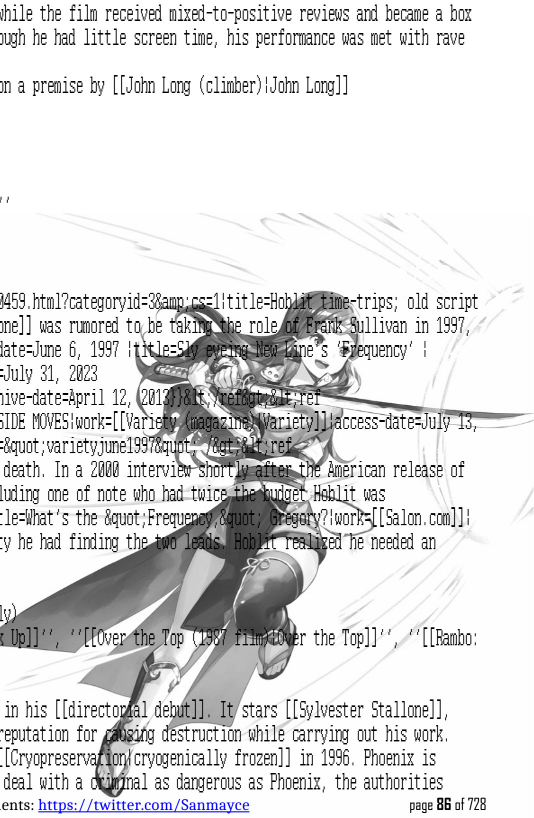
482 File:Sylvester Stallone Cannes 2019.jpg¦[[Sylvester Stallone]], Worst Actor of the Decade winner.

483 ¦[[Sylvester Stallone]] in ''[[Lock Up (1989 film)¦Lock Up]]'' and ''[[Tango &amp; Cash]]'' as Frank Leone and Ray Tango (respectively)

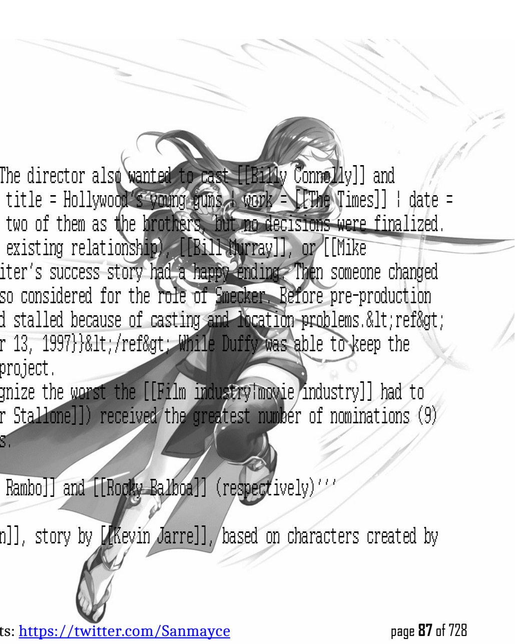
484 ¦ style=&quot;background:#B0C4DE;&quot; ¦'''[[Sylvester Stallone]], for ''[[Cobra (1986 film)¦Cobra]]'', ''[[Lock Up (1989 film)¦Lock Up]]'', ''[[Over the Top (1987 film)¦Over the Top]]'', ''[[Rambo: First Blood Part II]]'', ''[[Rambo III]]'', ''[[Rhinestone (film)¦Rhinestone]]'', ''[[Rocky IV]]'' and ''[[Tango &amp; Cash]]'''''

485 * [[Sylvester Stallone]]

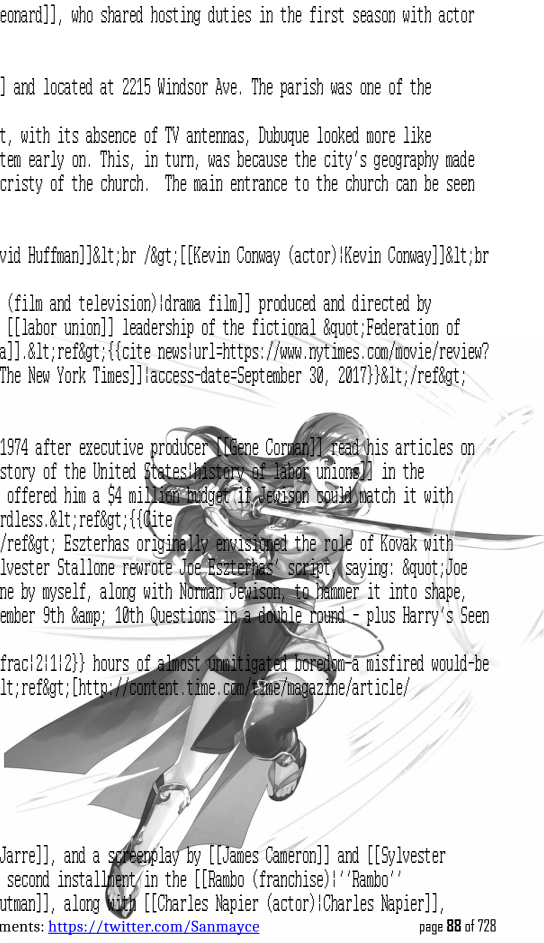
486 '''''Demolition Man''''' is a 1993 American [[Science fiction film¦science fiction]] [[action film]] directed by [[Marco Brambilla]] in his [[directorial debut]]. It stars [[Sylvester Stallone]], [[Wesley Snipes]], [[Sandra Bullock]], and [[Nigel Hawthorne]]. Stallone plays John Spartan, a risk-taking police officer who has a reputation for causing destruction while carrying out his work. After a failed attempt to rescue hostages from evil [[Crime boss¦crime lord]] Simon Phoenix (Snipes), they are both sentenced to be [[Cryopreservation¦cryogenically frozen]] in 1996. Phoenix is thawed for a parole hearing in 2032, but escapes. Society has changed and all violent crime has seemingly been eliminated. Unable to deal with a criminal as dangerous as Phoenix, the authorities



```
    awaken Spartan to help capture him again. The story makes allusions to many other works including [[Aldous Huxley]]'s 1932 [[Utopian and dystopian fiction|dystopian novel]] ''[[Brave New
    World]]''&lt;ref name=&quot;James&quot; /&gt; and [[H. G. Wells]]'s ''[[The Sleeper Awakes]]''.&lt;ref name=&quot;Lambie&quot;&gt;{{cite web |date=20 November 2016 |first=Ryan |last=Lambie |
    title=Demolition Man: It's 20 Years Since Stallone Was Frozen |url=https://www.denofgeek.com/movies/demolition-man-its-20-years-since-stallone-was-frozen/ |website=[[Den of Geek]] }}&lt;/ref&gt;
487 | image1           = Sylvester Stallone Comic-Con 2010.jpg
488 | alt1             = Sylvester Stallone
489 | link1            = File:Sylvester Stallone Comic-Con 2010.jpg
490 * [[Sylvester Stallone]] as John Spartan
491 * [[Nigel Hawthorne]] as Dr. Raymond Cocteau&lt;ref&gt;{{cite web |date=8 January 1995 |first=David |last=Gritten |title=Late-Blooming Nigel Hawthorne Enjoys 'Madness' of King-Size Role in Hytner's
    Film |url=https://www.latimes.com/archives/la-xpm-1995-01-08-ca-17652-story.html |work=Los Angeles Times |quote=Nor did he enjoy his role in &quot;Demolition Man&quot; (1993), with Sylvester
    Stallone, which he has never seen. }}&lt;/ref&gt;&lt;ref&gt;{{cite web |date=January 22, 2003 |last=Taylor |first=Markland |title=Straight Face, The Autobiography
    |url=https://variety.com/2003/more/reviews/straight-face-the-autobiography-1200543941/ |website=Variety |quote=referring to the experience as &quot;miserable&quot; as the two thoughtless stars kept
    everyone waiting. }}&lt;/ref&gt;
492 &quot;San Angeles&quot; was filmed in Orange County, California. Several locations in Irvine and San Diego were also used.&lt;ref name=&quot;Hayner&quot; /&gt;&lt;ref name=&quot;Wong&quot;&gt;{{cite
    news |title=Irvine Cast as Futuristic L.A. : Movie: Action-thriller starring Wesley Snipes and Sylvester Stallone is being filmed in the city this week. |newspaper=Los Angeles Times |date=April 16,
    1993 |url=https://www.latimes.com/archives/la-xpm-1993-04-16-me-23636-story.html |access-date=November 8, 2010 |first=Stacy |last=Wong }}&lt;/ref&gt;&lt;ref&gt;{{cite web |title=Filming Locations for
    Demolition Man (1993) |url=https://www.movie-locations.com/movies/d/Demolition-Man.php |website=The Worldwide Guide to Movie Locations }}&lt;/ref&gt;
493 A subplot involving Spartan's daughter was cut for pacing reasons. This led to some confusion at test screenings, where audiences thought Sandra Bullock was the daughter, and reacted negatively to
    the scene where they were about to have sex.&lt;ref name=&quot;Willmore&quot; /&gt; Originally Spartan's daughter was one of the Scraps living underground with Edgar Friendly's resistance.&lt;ref
    name=&quot;Hayner&quot; /&gt; A scene where Stallone fights Jesse Ventura was cut from the film.&lt;ref&gt;{{cite web |title=(VIDEO) Stallone/Ventura Talk Demolition Man Fight Scene
    |url=http://www.manlymovie.net/2015/02/video-stalloneventura-talk-demolition-man-fight-scene.html |website=ManlyMovie.net}} {{cite AV media |author=Top Turnbuckle |title=Eric Bischoff interviews
    Sylvester Stallone and Jesse Ventura (05-22-1993) |url=https://www.youtube.com/watch?v=9kZ5OeMEdKM&amp;feature=youtu.be&amp;t=270 |
    archive-url=https://web.archive.org/web/20200814203106/https://www.youtube.com/watch?v=9kZ5OeMEdKM&amp;gl=US&amp;hl=en| archive-date=2020-08-14 | url-status=dead |website=[[YouTube]] |date=Jan 16,
    2017 }}&lt;/ref&gt;{{better source needed|date=March 2023}}
494 The theme song &quot;[[Love Boat (song)|Love Boat]]&quot; of the [[The&amp;nbsp;Love&amp;nbsp;Boat|series of the same name]], which is also not part of the soundtrack, was played by Sandra Bullock as
    a romantic background music in her apartment before the &quot;virtual sex&quot; between her and Sylvester Stallone.&lt;ref name=&quot;James&quot;/&gt;
495 In 2017, [[Sylvester Stallone]]'s [[Loan-out corporation|loan-out company]] filed a lawsuit against [[Warner Bros.]] over the disbursement of profits from the film.&lt;ref&gt;{{cite web |date=April
    13, 2017 |last=Mumford |first=Gwilym |title=Sylvester Stallone sues Warner Bros for 'dishonesty' over Demolition Man profits |url=https://www.theguardian.com/film/2017/apr/13/sylvester-stallone-sues-
    warner-bros-demolition-man |website=[[The Guardian]] |url-status=live |archive-url=https://web.archive.org/web/20170413100809/https://www.theguardian.com/film/2017/apr/13/sylvester-stallone-sues-
    warner-bros-demolition-man |archive-date=2017-04-13 |access-date=July 11, 2020 }}&lt;/ref&gt;&lt;ref&gt;{{cite web |date=13 April 2017 |title=Sylvester Stallone sues 'greedy' studio
    |url=https://www.bbc.com/news/entertainment-arts-39587583 |website=BBC News }}&lt;/ref&gt; The lawsuit was settled in 2019.&lt;ref&gt;{{cite web |date= May 8, 2019 |last= Maddaus |first= Gene |title=
    Sylvester Stallone Settles 'Demolition Man' Profits Dispute |url= https://variety.com/2019/biz/news/sylvester-stallone-demolition-man-settlement-1203208843/ |website=Variety |url-status=live |
    archive-url=https://web.archive.org/web/20190508195041/https://variety.com/2019/biz/news/sylvester-stallone-demolition-man-settlement-1203208843/ |archive-date= 2019-05-08 }}&lt;/ref&gt;
496 On [[Rotten Tomatoes]], the film has an approval rating of 63% based on 43 reviews, with an average rating of 5.60/10. The site's consensus reads: &quot;A better-than-average sci-fi shoot-em-up with
    a satirical undercurrent, ''Demolition Man'' is bolstered by strong performances by Sylvester Stallone, Wesley Snipes, and Sandra Bullock.&quot;&lt;ref&gt;{{cite web |title=Demolition Man (1993) |
    url=https://www.rottentomatoes.com/m/demolition_man/ |website=[[Rotten Tomatoes]] |publisher= Fandango Media |access-date=February 1, 2024 }}&lt;/ref&gt; On [[Metacritic]] the film has a [[weighted
    arithmetic mean|weighted average]] score of 34 out of 100, based on nine reviews, indicating &quot;generally unfavorable reviews&quot;.&lt;ref&gt;{{cite web |title=Demolition Man Reviews |
    url=https://www.metacritic.com/movie/demolition-man |website=[[Metacritic]] |publisher= CBS |access-date=July 31, 2020 }}&lt;/ref&gt; Audiences polled by [[CinemaScore]] gave the film an average
    grade of &quot;B&quot; on an A+ to F scale.&lt;ref name=&quot;CinemaScore&quot;&gt;{{cite web |url= https://www.cinemascore.com/publicsearch/index/title/ |title= DEMOLITION MAN (1993) B |work=
    [[CinemaScore]] |url-status= dead |archive-url= https://web.archive.org/web/20180206073531/https://www.cinemascore.com/publicsearch/index/title/ |archive-date= February 6, 2018 }}&lt;/ref&gt;
497 Demolition Man action figures and vehicles were released in 1993. Produced by Mattel the toys were based on their &quot;[[The New Adventures of He-Man|New Adventures of He-Man]]&quot; style of
    figures. In addition to seven action figure, the set included a car, a red convertible called the &quot;Fast Blast 442&quot;&lt;!-- previously released in yellow as Last Action Hero Slater's
    convertible --&gt;, an airplane &quot;Bolajet&quot; &lt;!-- previously released in 1989 as He-Man Bolajet https://www.he-man.org/collecting/toy.php?id=1343 --&gt;, and a &quot;Missile Shooter&quot;
    toy gun.&lt;ref&gt;{{cite web |title=Demolition Man (Mattel) Action Figure Checklist |url=https://www.figurerealm.com/actionfigure?action=seriesitemlist&amp;id=300
    |website=FigureRealm.com}}&lt;/ref&gt; Lenina Huxley was not included in the toy line.&lt;ref&gt;{{cite web |date=24 November 2014 |first1=Michael |last1=Roffman |first2=Dan |last2=Caffreyon |
    title=The 5 Worst and Best Movie Action Figures |url=https://consequenceofsound.net/2014/11/worst-and-best-movie-action-figures/5/ |website=Consequence of Sound }}&lt;/ref&gt;&lt;ref&gt;{{cite web |
    date=2013-03-01 |author=MTVGEEK |title=The Many (Toy) Faces of Sylvester Stallone |url=http://www.mtv.com/news/2628210/sylvester-stallone-toys/
    |archive-url=https://web.archive.org/web/20160514184315/http://www.mtv.com/news/2628210/sylvester-stallone-toys/ |url-status=dead |archive-date=May 14, 2016 |website=[[MTV News]] |quote=While Snipes
    got his first figure with the line, sadly Sandra Bullock's character received no figure as well. }}&lt;/ref&gt;
498 [[Acclaim Entertainment]] and [[Virgin Interactive]] released ''[[Demolition Man (video game)|Demolition Man]]'' on various home video game systems. The 16-bit versions were shooting games
    distributed by Acclaim. The [[3DO Interactive Multiplayer|3DO]] version is a multi-genre game that incorporates [[Full Motion Video]] scenes, with both [[Sylvester Stallone]] and [[Wesley Snipes]]
    reprising their roles as their characters in scenes that were filmed exclusively for the game.&lt;ref&gt;{{cite magazine |title=Demolition Man |magazine=[[GamePro]] |issue=76 |
    publisher=[[International Data Group|IDG]] |date=January 1995 |page=192 }}&lt;/ref&gt;
499 The film found renewed relevance during the [[COVID-19]] pandemic; the film was seen as predictive when there were calls to end the practice of shaking hands, and shortages of toilet paper.&lt;ref
    name=&quot;Willmore&quot; /&gt;&lt;ref&gt;{{cite web |date=6 April 2020 |last=Augustine |first=Afiya |title=Are we headed towards a Demolition Man future? |url=https://www.syfy.com/syfywire/are-we-
    headed-towards-a-demolition-man-future |website=SYFY WIRE }}&lt;/ref&gt;&lt;ref name=&quot;Collis&quot;&gt;{{cite magazine |date= May 4, 2020 |first= Clark |last= Collis |title=Sylvester Stallone is
    'working on' a sequel to 'Demolition Man' |url=https://ew.com/movies/demolition-man-sequel-sylvester-stallone/ |magazine=Entertainment Weekly |quote=The film recently resurfaced in the popular
    consciousness when there were concerns of a toilet paper shortage following the outbreak of the coronavirus pandemic. }}&lt;/ref&gt;&lt;ref&gt;{{cite web |date=20 April 2020 |first=Jon |last=Fuge |
    title='Demolition Man' Writer Looks at How It Predicted the Future and a Potential Sequel |url=https://movieweb.com/demolition-man-sequel-future-predictions/ |website=Movieweb
    }}&lt;/ref&gt;&lt;ref&gt;{{cite web |date=2020-04-16 |first=Beth |last=Elderkin |title=Demolition Man's Writer Wasn't Trying to Be Prescient, He Just Wanted to Make a Funny Movie
    |url=https://io9.gizmodo.com/demolition-mans-writer-wasnt-trying-to-be-prescient-he-1842901085 |website=io9 }}&lt;/ref&gt;
500 In 2022, Sylvester Stallone reflected positively on the movie: &quot;I always enjoyed this movie. It was a great action film wonderfully directed by Marco Brambilla. And the writers were way ahead of
    their time.&quot;&lt;ref&gt;{{cite web |date=13 February 2022 |last1=Dick |first1=Jeremy |title=Sylvester Stallone Reflects on Demolition Man, Says It Was Ahead of Its Time
    |url=https://movieweb.com/sylvester-stallone-reflects-demolition-man/ |website=MovieWeb.com }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://www.instagram.com/p/CZ7KXcBJJcF/ |
    title=Instagram }}&lt;/ref&gt;
501 File:Sylvester Stallone Cannes 2019.jpg|[[Sylvester Stallone]], Worst Actor winner.
502 | style=&quot;background:#B0C4DE;&quot; |'''[[Sylvester Stallone]] in ''[[Rambo III]]'' as [[John Rambo]]'''
503 |''[[Rambo III]]'', written by [[Sylvester Stallone]] and [[Sheldon Lettich]], based on characters created by [[David Morrell]]
504 |[[Sylvester Stallone]] in ''[[Over the Top (1987 film)|Over the Top]]'' as Lincoln Hawk
505 |[[Sylvester Stallone]] in ''[[Cobra (1986 film)|Cobra]]'' as Marion &quot;Cobra&quot; Cobretti
506 |''[[Cobra (1986 film)|Cobra]]'', screenplay by [[Sylvester Stallone]], based on the novel ''Fair Game'' by [[Paula Gosling]]
507 Duffy sought to cast [[Stephen Dorff]] and [[Mark Wahlberg]] as the brothers, though Wahlberg passed for ''[[Boogie Nights]]'' (1997). The director also wanted to cast [[Billy Connolly]] and
    [[Kenneth Branagh]] in the film, with Branagh playing FBI Agent Paul Smecker.&lt;ref&gt;{{cite news | last = O'Toole | first = Lesley | title = Hollywood's young guns | work = [[The Times]] | date =
    December 4, 1997}}&lt;/ref&gt; Duffy also expressed interest in casting [[Brendan Fraser]], [[Nicky Katt]], and [[Ewan McGregor]], with two of them as the brothers, but no decisions were finalized.
    The director later sought [[Patrick Swayze]] to play Smecker, but Miramax preferred [[Sylvester Stallone]] (with whom the studio had an existing relationship), [[Bill Murray]], or [[Mike
    Myers]].&lt;ref name=&quot;faces&quot;&gt;{{cite news | last = Waxman | first = Sharon | title = The two faces of Hollywood: A screenwriter's success story had a happy ending. Then someone changed
    the script. | newspaper = [[The Washington Post]] | date = April 10, 1998}}&lt;/ref&gt; [[Kevin Spacey]] and [[Robert De Niro]] were also considered for the role of Smecker. Before pre-production
    work was supposed to begin in Boston in December 1997, Miramax pulled out of the project. Producer Lloyd Segan said that the project had stalled because of casting and location problems.&lt;ref&gt;
    {{cite news | last = Raposa | first = Laura | title = Miramax 'Saints' hits devil of a snag | work = [[Boston Herald]] | date = December 13, 1997}}&lt;/ref&gt; While Duffy was able to keep the
    writer's fee of $300,000, the studio required the reimbursement of the $150,000 director's fee and the $700,000 advance to develop the project.
508 The '''6th Golden Raspberry Awards''' were held on March 23, 1986, at the Morgan-Wixon Theatre in [[Santa Monica, California]], to recognize the worst the [[Film industry|movie industry]] had to
    offer in [[1985 in film|1985]]. Though ''[[Rambo: First Blood Part II]]'' won Worst Picture, ''[[Rocky IV]]'' (also starring [[Sylvester Stallone]]) received the greatest number of nominations (9)
    and &quot;wins&quot; (5). This is the only year in which one movie won worst picture and another movie had the most nominations and wins.
509 File:Sylvester Stallone Cannes 2019.jpg|[[Sylvester Stallone]], Worst Director and Worst Actor winner, Worst Screenplay co-winner.
510 | style=&quot;background:#B0C4DE;&quot; |'''[[Sylvester Stallone]] in ''[[Rambo: First Blood Part II]]'' and ''[[Rocky IV]]'' as [[John Rambo]] and [[Rocky Balboa]] (respectively)'''
511 | style=&quot;background:#B0C4DE;&quot; |'''[[Sylvester Stallone]] for ''[[Rocky IV]]'''''
512 | style=&quot;background:#B0C4DE;&quot; |''''[[Rambo: First Blood Part II]]'', screenplay by [[Sylvester Stallone]] and [[James Cameron]], story by [[Kevin Jarre]], based on characters created by
    [[David Morrell]]'''
513 |''[[Rocky IV]]'', written by [[Sylvester Stallone]]
514 |[[Sylvester Stallone]] in ''[[Daylight (1996 film)|Daylight]]'' as Kit Latura
```



```
515 Morgan began to work as a freelance at ''[[The Sun (United Kingdom)|The Sun]]'' in 1988, at this point dropping his [[double-barrelled name]]. He told [[Hunter Davies]] in December 1994 that he was personally recruited by ''Sun'' editor [[Kelvin MacKenzie]] to work on the newspaper's show business column &quot;Bizarre&quot;, his first high-profile post.&lt;ref name=&quot;Davies1994&quot;/&gt; Although he was not a fan of pop music, he was considered skilled at self-publicity and became the column's main writer. &quot;I became the Friend of the Stars, a rampant egomaniac, pictured all the time with famous people - [[Madonna (entertainer)|Madonna]], [[Sylvester Stallone|Stallone]], [[David Bowie|Bowie]], [[Paul McCartney]], hundreds of them. It was shameless, as they didn't know me from Adam&quot;, he told Davies.&lt;ref name=&quot;Davies1994&quot;/&gt;
516 |[[Sylvester Stallone]] in ''[[Assassins (1995 film)|Assassins]]'' and ''[[Judge Dredd (film)|Judge Dredd]]'' as Robert Rath and [[Judge Joseph Dredd]] (respectively)
517 File:Sylvester Stallone 2012.jpg|[[Sylvester Stallone]], Worst Screen Couple co-winner.
518 |[[Sylvester Stallone]] in ''[[The Specialist]]'' as Ray Quick
519 | style=&quot;background:#B0C4DE;&quot; |'''[[Sylvester Stallone]] and [[Sharon Stone]] in ''[[The Specialist]]'' (tie)'''
520 |[[Sylvester Stallone]] (as himself) in ''[[An Alan Smithee Film: Burn Hollywood Burn]]''
521 [[File:Sylvester Stallone 2012.jpg|thumb|right|150px|[[Sylvester Stallone]], Worst Actor winner]]
522 *'''[[Sylvester Stallone]]&amp;nbsp;- ''[[Rhinestone (film)|Rhinestone]]'' as Nick Martinelli'''
523 **''[[Rhinestone (film)|Rhinestone]]''&amp;nbsp;- screenplay by [[Phil Alden Robinson]] and [[Sylvester Stallone]], story by Robinson
524 Snipes has played a number of roles in action films, including ''[[Passenger 57]]'', ''[[Rising Sun (film)|Rising Sun]]'', ''[[Demolition Man (film)|Demolition Man]]'' (with [[Sylvester Stallone]]), ''[[Drop Zone (film)|Drop Zone]]'', ''[[Money Train]]'', ''[[The Fan (1996 film)|The Fan]]'', ''[[U.S. Marshals (film)|U.S. Marshals]]'' and ''[[The Art of War (film)|The Art of War]]'', as well as comedies like ''[[White Men Can't Jump]]'', and ''[[To Wong Foo, Thanks for Everything! Julie Newmar]]'' where he played a [[drag queen]]. Snipes has appeared in additional dramas, such as ''[[The Waterdance]]'' and ''[[Disappearing Acts]]''.&lt;ref&gt;{{cite web|url=http://www.biography.com/people/wesley-snipes-9542590#film-debut-wildcats|title=Wesley Snipes|website=Biography.com|access-date=December 11, 2017}}&lt;/ref&gt;
525 |[[Sylvester Stallone]] in ''[[Get Carter (2000 film)|Get Carter]]'' as Jack Carter
526 [[Sylvester Stallone]] was originally considered for the role of Jack T. Colton.&lt;ref&gt;Plumb, Ali. [http://www.empireonline.com/features/evening-with-sylvester-stallone-quotes/p1 &quot;14 things we learned from the 'Evening with Sylvester Stallone' Q&amp;A.&quot;] ''[[Empire (film magazine)|Empire Online]]'', October 9, 2015. Retrieved: March 28, 2016.&lt;/ref&gt; Other leads considered include [[Burt Reynolds]], [[Clint Eastwood]], [[Paul Newman]], and [[Christopher Reeve]] for the part of Jack Colton before Douglas hired himself and [[Debra Winger]] was the studio's top choice for Joan Wilder.&lt;ref&gt;Susman, Gary. [http://www.moviefone.com/2014/03/24/romancing-the-stone-facts/ &quot;'Romancing the Stone': 25 things you didn't know about the Kathleen Turner classic.&quot;] ''moviefone.com'', March 24, 2014. Retrieved: March 28, 2016.&lt;/ref&gt;&lt;ref&gt;Leigh 2014, p. 171.&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=King |first=Susan |date=2019-03-30 |title='Romancing the Stone' at 35: How Michael Douglas, Kathleen Turner Survived Gators, Rain, and Studio Expectations |url=https://variety.com/2019/film/features/romancing-the-stone-35-anniversary-michael-douglas-1203175725/ |access-date=2024-06-28 |website=Variety |language=en-US}}&lt;/ref&gt;
527 * [[Sylvester Stallone]]
528 Meyer moved to Los Angeles in 1993 to appear in the Fox teen drama series ''[[Beverly Hills, 90210]]'' in the recurring role of Lucinda Nicholson. Shortly afterwards, she was cast as the female lead in the action film ''[[Johnny Mnemonic (film)|Johnny Mnemonic]]'' opposite [[Keanu Reeves]]. The following year, she appeared in the fantasy film ''[[Dragonheart]]'' directed by [[Rob Cohen]]. In early 1997, Meyer had a recurring role as Kate Miller in the NBC sitcom ''[[Friends]]''. Later that year, she went to appear alongside [[Casper Van Dien]] in the science-fiction film ''[[Starship Troopers (film)|Starship Troopers]]'', directed by [[Paul Verhoeven]]. The film underperformed at the box office.&lt;ref&gt;{{Cite web|title=Starship Troopers|url=https://www.boxofficemojo.com/release/rl3144910337/|access-date=2020-08-06|website=Box Office Mojo}}&lt;/ref&gt; The following year, she starred alongside [[James Caan]] in the neo-noir film ''[[Poodle Springs (film)|Poodle Springs]]'', which was released on [[HBO]]. In 1999, she starred in the horror film ''[[Bats (film)|Bats]]''; the film was a moderate box-office success.&lt;ref&gt;{{Cite web|title=Bats|url=https://www.boxofficemojo.com/release/rl3359999489/weekend/|access-date=2020-08-06|website=Box Office Mojo}}&lt;/ref&gt; In 2002, she appeared opposite [[Sylvester Stallone]] in the thriller film ''[[D-Tox]]''. Also in 2002, Meyer appeared as Romulan Commander Donatra in the science-fiction film ''[[Star Trek: Nemesis]]''.
529 |&lt;ref&gt;{{Cite web |last=McCall |first=Kevin |date=2023-04-12 |title=Sylvester Stallone and Nathan Fillion Hijack the New 'Guardians of the Galaxy Vol. 3' Trailer |url=https://collider.com/guardians-of-the-galaxy-vol-3-trailer-sylvester-stallone-nathan-fillion/ |access-date=2024-07-27 |website=Collider |language=en}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Freitag |first=Lee |date=2023-04-27 |title=GOTG Vol. 3 Images Debut New Look at Nathan Fillion's Mystery Alien |url=https://www.cbr.com/guardians-of-the-galaxy-vol-3-nathan-fillion-image/ |access-date=2024-07-27 |website=CBR |language=en}}&lt;/ref&gt;
530 Scott married three times. His first marriage was to TV production designer [[Gerry Scott|Gerry Boldy]] (1944-2007) in 1967; they were divorced in 1974.&lt;ref name=gerryscott&gt;{{cite news|last=Stafford-Clark|first=Nigel|title=Obituary: Gerry Scott Foulds|url=https://www.theguardian.com/media/2007/may/12/broadcasting.guardianobituaries|access-date=27 August 2012|newspaper=The Guardian|date=12 May 2007|location=London}}&lt;/ref&gt; His second marriage was in 1986 to advertising executive Glynis Sanders;&lt;ref&gt;{{cite news|last1=Hough|first1=Andrew|last2=Allen|first2=Nick|url=https://www.telegraph.co.uk/news/celebritynews/9486673/Top-Gun-director-Tony-Scott-dies-after-jumping-from-Los-Angeles-bridge.html |archive-url=https://ghostarchive.org/archive/20220111/https://www.telegraph.co.uk/news/celebritynews/9486673/Top-Gun-director-Tony-Scott-dies-after-jumping-from-Los-Angeles-bridge.html |archive-date=11 January 2022 |url-access=subscription |url-status=live|title=Top Gun director Ton y Scott dies after jumping from Los Angeles bridge|work=The Telegraph|date=20 August 2012|access-date=16 October 2012|location=London}}{{cbignore}}&lt;/ref&gt; they divorced a year later when his affair with [[Brigitte Nielsen]] (married to [[Sylvester Stallone]] at the time), whom he met on the set of ''Beverly Hills Cop II'', became public.{{citation needed|date=December 2018}} He subsequently met film and TV actress Donna Wilson, who was 24 years his junior, on the set of ''Days of Thunder'' in 1990; they married in 1994. She gave birth to their twin sons, Frank and Max, in 2000.&lt;ref name=eveningstandard&gt;{{cite news|title=Hollywood pays tribute to Top Gun director Tony Scott following suicide leap|url=https://www.standard.co.uk/showbiz/celebrity-news/hollywood-pays-tribute-to-top-gun-director-tony-scott-following-suicide-leap-8062363.html|access-date=27 August 2012|newspaper=London Evening Standard|date=20 August 2012}}&lt;/ref&gt;
531 * [[Sylvester Stallone]]
532 The series was created and executive produced by [[Mark Burnett]]. The first three seasons of the show were hosted by [[Sugar Ray Leonard]], who shared hosting duties in the first season with actor [[Sylvester Stallone]]. Leonard departed the series after season three and [[Tony Danza]] joined as his replacement.
533 |[[Sylvester Stallone]]&lt;br /&gt;[[Sugar Ray Leonard]]
534 '''Sacred Heart Catholic Church''' is a [[Roman Catholic]] church in [[Dubuque, Iowa]]. It is part of the [[Archdiocese of Dubuque]] and located at 2215 Windsor Ave. The parish was one of the locations where the movie ''[[F.I.S.T. (film)|F.I.S.T.]]'' starring [[Sylvester Stallone]] was filmed.
535 The movie ''[[F.I.S.T. (film)|F.I.S.T.]]'', starring [[Sylvester Stallone]], was filmed largely in Dubuque. The main reason was that, with its absence of TV antennas, Dubuque looked more like [[Cleveland, Ohio|Cleveland]] of the 1930s than Cleveland did. There aren't many antennas because Dubuque had developed a cable system early on. This, in turn, was because the city's geography made over-the-air reception problematic. The church basement was one of the many Dubuque locations where filming was done, as was the sacristy of the church.  The main entrance to the church can be seen in the wedding scenes that were filmed at the church.
536 | screenplay  = [[Joe Eszterhas]]&lt;br /&gt;[[Sylvester Stallone]]
537 | starring = Sylvester Stallone&lt;br /&gt;[[Rod Steiger]]&lt;br /&gt;[[Peter Boyle]]&lt;br /&gt;[[Melinda Dillon]]&lt;br /&gt;[[David Huffman]]&lt;br /&gt;[[Kevin Conway (actor)|Kevin Conway]]&lt;br /&gt;[[Tony Lo Bianco]]
538 '''''F.I.S.T.''''' (stylized on-screen as '''''F•I•S•T''''') is a 1978 American [[Action film|action]] [[crime film|crime]] [[Drama (film and television)|drama film]] produced and directed by [[Norman Jewison]] and starring [[Sylvester Stallone]]. Stallone plays a [[Cleveland]] warehouse worker who becomes involved in the [[labor union]] leadership of the fictional &quot;Federation of Inter-State Truckers&quot; (F.I.S.T.). The film is loosely based on the [[Teamsters Union]] and their former President [[Jimmy Hoffa]].&lt;ref&gt;{{cite news|url=https://www.nytimes.com/movie/review?res=9904E0DB1231E632A25755C2A9629C946990D6CF&amp;mcubz=0|title=Screen: 'F.I.S.T.,' Drama of Unionism: Stallone Returns|newspaper=[[The New York Times]]|access-date=September 30, 2017}}&lt;/ref&gt; The screenplay was by [[Joe Eszterhas]] and Stallone, from a story by Eszterhas.
539 * [[Sylvester Stallone]] as Johnny Kovak
540 [[United Artists]] hired [[Joe Eszterhas]] to write a screenplay about the [[Labor unions in the United States|labor movement]] in 1974 after executive producer [[Gene Corman]] read his articles on the subject in ''[[Rolling Stone]]''. Eszterhas, who had never written a screenplay before, wrote a 40-page-essay on the [[Labor history of the United States|history of labor unions]] in the [[Midwestern United States|American Midwest]]. [[Arthur B. Krim]] hired [[Norman Jewison]] in the position of producer-director and offered him a $4 million budget if Jewison could match it with another $4 million. He eventually failed to find another studio to co-finance the film and received a full $8.1 million budget regardless.&lt;ref&gt;{{Cite web|title=F.I.S.T.|url=https://catalog.afi.com/Film/55253-FIST?cxt=filmography|access-date=2022-02-17|website=catalog.afi.com}}&lt;/ref&gt; Eszterhas originally envisioned the role of Kovak with [[Jack Nicholson]] (who would later play Jimmy Hoffa, on whom Kovak was loosely based), but Sylvester Stallone was cast instead. Sylvester Stallone rewrote Joe Eszterhas' script, saying: &quot;Joe Eszterhas wrote a script that was nearly 400 pages and was more of a novel than a shootable screenplay. A great deal of work was done by myself, along with Norman Jewison, to hammer it into shape, but Joe had conceived a great idea.&quot;&lt;ref&gt;{{cite web|url=https://www.aintitcool.com/node/30932|title=Stallone answers December 9th &amp; 10th Questions in a double round - plus Harry's Seen Rocky Balboa at BNAT!!! |date=December 16, 2006 |website=[[Ain't It Cool News]]|access-date=September 29, 2017}}&lt;/ref&gt;
541 A negative review came from [[Richard Schickel]] of ''[[Time (magazine)|Time]]'', who said that the film &quot;stands for nearly {{frac|2|1|2}} hours of almost unmitigated boredom-a misfired would-be proletarian epic with Sylvester Stallone misplaying the Jimmy Hoffa part with a self-confidence that borders on the sublime.&quot;&lt;ref&gt;[http://content.time.com/time/magazine/article/0,9171,919573,00.html Cinema: J.U.N.K.], ''[[Time (magazine)|Time]]'', May 1, 1978.&lt;/ref&gt;
542 {{Sylvester Stallone}}
543 [[Category:Films with screenplays by Sylvester Stallone]]
544 Da Matta and his 1999 car appear in the movie ''[[Driven (2001 film)|Driven]]'', starring [[Sylvester Stallone]].
545 | screenplay     = [[Sylvester Stallone]]&lt;br /&gt;[[James Cameron]]
546 * Sylvester Stallone
547 '''''Rambo: First Blood Part II''''' is a 1985 American [[action film]] directed by [[George P. Cosmatos]] from a story by [[Kevin Jarre]], and a screenplay by [[James Cameron]] and [[Sylvester Stallone]], who also reprises his role as [[Vietnam War]] veteran [[John Rambo]]. A sequel to ''[[First Blood]]'' (1982), it is the second installment in the [[Rambo (franchise)|''Rambo'' franchise]], followed by ''[[Rambo III]]''. It co-stars [[Richard Crenna]], who reprises his role as [[Sam Trautman|Colonel Sam Trautman]], along with [[Charles Napier (actor)|Charles Napier]],
```



[[Julia Nickson]], and [[Steven Berkoff]].

548 * [[Sylvester Stallone]] as [[John Rambo|John J. Rambo]]

549 [[Vincent Canby]] of ''[[The New York Times]]'' called the film &quot;almost as opportunistic as the [[Member of Congress|Congressman]] it pretends to abhor. In spite of everything it says, it's much less interested in the M.I.A. question than it is in finding a topical frame for the kind of action-adventure film in which Mr. Stallone - his torso and his vacant stare - can do what his fans like best. That is, fight, outwit and kill, usually all by himself, dozens of far-better armed but lesser mortals.&quot;&lt;ref&gt;Canby, Vincent (May 26, 1985). [https://www.nytimes.com/1985/05/26/movies/film-view-rambo-delivers-a-revenge-fantasy.html &quot;'Rambo' Delivers A Revenge Fantasy&quot;.] ''[[The New York Times]]''. H11.&lt;/ref&gt; ''[[Variety (magazine)|Variety]]'' wrote, &quot;The charade on the screen, which is not pulled off, is to accept that the underdog Rambo character, albeit with the machine-gun wielding help of an attractive Vietnamese girl, can waste hordes of [[Viet Cong]] and [[Soviet Army|Red Army]] contingents en route to hauling POWs to a Thai air base in a smoking Russian chopper with only a facial scar (from a branding iron-knifepoint) marring his tough figure. You never even see him eating in this fantasy, as if his body feeds on itself.&quot;&lt;ref&gt;&quot;Film Reviews: Rambo: First Blood Part II&quot;. ''[[Variety (magazine)|Variety]]''. May 22, 1985. 14.&lt;/ref&gt; [[Gene Siskel]] of the ''[[Chicago Tribune]]'' gave the film three stars out of four and called it &quot;very good at what it does, but what it does isn't always that good&quot;, referring to the depiction of the enemy as going &quot;back to the image of the [[Yellow Peril]], to the notion that white is right and other colors are wrong.&quot;&lt;ref&gt;Siskel, Gene (May 22, 1985). &quot;'Rambo': Cinematic soldiering whitewashes Vietnam&quot;. ''[[Chicago Tribune]]''. Section 5, p. 1, 3.&lt;/ref&gt; Michael Wilmington of the ''[[Los Angeles Times]]'' wrote, &quot;If a character can seemingly do anything, it's hard to feel tension or concern about his fate. (At least [[Superman]] had [[kryptonite]].) We are left with nothing but detached aesthetic appreciation: watching Rambo race through several million dollars worth of explosions and aerial attacks, coruscant fireballs billowing everywhere and bodies flying hither and yon. Except for anyone irretrievably into violent power fantasies, this will probably soon pall.&quot;&lt;ref&gt;Wilmington, Michael (May 22, 1985). [https://www.latimes.com/archives/la-xpm-1985-05-22-ca-16965-story.html &quot;Why a 'Rambo II'? For Muddiest of Reasons&quot;.] ''[[Los Angeles Times]]''. Part VI, p. 1, 6.&lt;/ref&gt; [[Pauline Kael]] commented in ''[[The New Yorker]]'', &quot;The director, George P. Costmatos, gives this near-psychotic material-a mixture of [[Catholic Church|Catholic]] iconography and ''[[Soldier of Fortune (magazine)|Soldier of Fortune]]'' pulp-a veneer of professionalism, but the looniness is always there.&quot;&lt;ref&gt;Kael, Pauline (June 17, 1985). &quot;The Current Cinema&quot;. ''[[The New Yorker]]''. 117.&lt;/ref&gt; [[Paul Attanasio]] of ''[[The Washington Post]]'' wrote, &quot;At best, ''Rambo: First Blood Part II'' is a crudely effective right-wing rabble-rouser, the artistic equivalent of carpet bombing-you don't know whether to cheer or run for cover. At worst, it's a tribute to Sylvester Stallone, by Sylvester Stallone, starring Sylvester Stallone.&quot;&lt;ref&gt;Attansasio, Paul (May 22, 1985). &quot;'Rambo': New Blood, Old Moves&quot;. ''[[The Washington Post]]''. F1.&lt;/ref&gt;

550 |rowspan=2|[[Sylvester Stallone]]

551 The 1986 arcade [[run and gun video game]] ''[[Ikari Warriors]]'' was intended by its developer [[SNK]] to be an official licensed adaptation of ''Rambo''. However, they were initially unable to acquire the rights to the film. This resulted in the game's title being changed to ''Ikari'', referencing part of the film's Japanese title, ''Rambo: Ikari no Dasshutsu'' (&quot;Rambo: The Furious Escape&quot;). After the game made its North American debut at an arcade game expo, they managed to get in touch with Sylvester Stallone about acquiring the rights to the film. However, it was too late by that point, as the game had already become popularly known by its Japanese ''Ikari'' title among arcade players in Japan and North America, which led to the game's official release as ''Ikari Warriors'' in North America. Stallone was friends with SNK's president at the time, and owned an ''Ikari Warriors'' [[arcade cabinet]].&lt;ref&gt;{{cite journal |title=「怒」を作った男 |trans-title=The Man Who Made &quot;Ikari&quot; |journal=[[:ja:CONTINUE (雑誌)|Continue]] |date=March 2001 |url=http://shmuplations.com/snkgoldenage/}}&lt;/ref&gt;

552 {{Sylvester Stallone}}

553 [[Category:Films with screenplays by Sylvester Stallone]]

554 1995 saw the release of ''[[Judge Dredd (film)|Judge Dredd]]'', a big budget version of the comic directed by [[Danny Cannon]] and starring [[Sylvester Stallone]]. Wagner was unhappy with the result, feeling they had filmed &quot;the wrong script&quot; and that &quot;Stallone was badly advised&quot;.&lt;ref name=Logan /&gt; A second attempt at adapting the character to the screen, entitled ''[[Dredd]]'', was released in September 2012, directed by [[Pete Travis]] from a script by [[Alex Garland]], and starring [[Karl Urban]]. This time Wagner was consulted over the script, was involved in the promotion of the film,&lt;ref&gt;Owen Williams, [https://www.empireonline.com/interviews/interview.asp?IID=1530 Exclusive: John Wagner And Alex Garland Talk Dredd], ''Empire'', 2012&lt;/ref&gt; and has described it as &quot;unlike the first film, a true representation of Judge Dredd&quot;.&lt;ref name=cbr&gt;Karl Keily, [http://www.comicbookresources.com/?page=article&amp;id=40172 &quot;John Wagner discusses 35 years of Judge Dredd&quot;], Comic Book Resources, 31 July 2012&lt;/ref&gt;

555 * [[Assassins (1995 film)|''Assassins'' (1995 film)]], starring Sylvester Stallone and Antonio Banderas

556 ''Face/Off'' was written by [[Mike Werb]] and [[Michael Colleary]] in 1990 as a [[spec script]]. They were inspired to write it after a mutual friend injured themself in a hang-gliding accident. The injury required the friend's face to be removed and reconstructed before being re-added onto their body. Werb and Colleary were also inspired by the film ''[[White Heat]]'' to create a plot where the main character survived a prison riot. ''Face/Off'' initially had a futuristic setting.&lt;ref name=&quot;:0&quot; /&gt;&lt;ref&gt;{{Cite web |date=1997 |title=Interviews: Produced Writer Mike Werb |url=http://www.writersupercenter.com/studionotes/interviews8.shtml |access-date=2024-02-27 |website=Writer Super Center}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Housman |first=Andrew |date=2022-03-23 |title=Believe It Or Not, Face/Off Could Have Been Even Weirder |url=https://www.slashfilm.com/807945/believe-it-or-not-faceoff-could-have-been-even-weirder/ |access-date=2024-02-27 |website=SlashFilm |language=en-US}}&lt;/ref&gt; ''Face/Off'' was optioned to [[Joel Silver]] and [[Warner Bros.]] in 1991. The option expired in 1994 and the project was purchased by [[Paramount Pictures]]. American director and producer [[Rob Cohen]] was originally set to direct the film but when the project was in a [[Turnaround (filmmaking)|turnaround]] Cohen left to direct ''[[Dragonheart]]'' (1996).&lt;ref&gt;{{Cite web |date=June 27, 2017 |title=15 Peachy Facts About Face/Off |url=https://www.mentalfloss.com/article/65363/15-peachy-facts-about-faceoff |website=Mental Floss}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Jones |first=Ralph |date=June 27, 2022 |title=The oral history of 'Face/Off,' Nicolas Cage's inexplicable sci-fi masterpiece |url=https://www.inverse.com/entertainment/face-off-oral-history/ |website=Inverse}}&lt;/ref&gt; [[John Woo]] became attached in 1996.&lt;ref name=&quot;Christopher Heard 2000&quot;&gt;Christopher Heard. ''Ten thousand bullets: the cinema of John Woo''. Los Angeles: Lone Eagle Publ, 2000. {{ISBN|1-58065-021-X}}&lt;/ref&gt; Various actor pairings were considered for the parts of Sean Archer and Castor Troy, such as [[Sylvester Stallone]] and [[Arnold Schwarzenegger]], [[Michael Douglas]] and [[Harrison Ford]], and [[Alec Baldwin]] and [[Bruce Willis]].&lt;ref&gt;{{cite book|last=Phipps|first= Keith |author-link= |date= March 29, 2022 |title= Age of Cage:Four Decades of Hollywood Through One Singular Career|url= https://books.google.com/books?id=7yIQEAAAQBAJ |publisher= Henry Holt and Company|chapter=8: Action Cage |isbn=9781250773036|quote=As various filmmakers considered the film, pairings like Sylvester Stallone and Arnold Schwarzenegger, Michael Douglas and Harrison Ford, and Alec Baldwin and Bruce Willis, came and went.}}&lt;/ref&gt;

557 Celebrity guests in attendance for WrestleMania 21 included [[David Arquette]], [[Motörhead]], [[Adam Sandler]], [[Rob Schneider]], [[Billy Gibbons]] from [[ZZ Top]], [[The Black Eyed Peas]], [[Billy Corgan]] from [[The Smashing Pumpkins]], [[Anthony Kiedis]] from [[Red Hot Chili Peppers]], [[Ice Cube]], [[Carmen Electra]], [[Sylvester Stallone]], [[Marg Helgenberger]], [[Matt Groening]], [[Rick Rubin]], [[Will Sasso]], and [[Macaulay Culkin]].&lt;ref name=&quot;SLAM! Review&quot;/&gt;

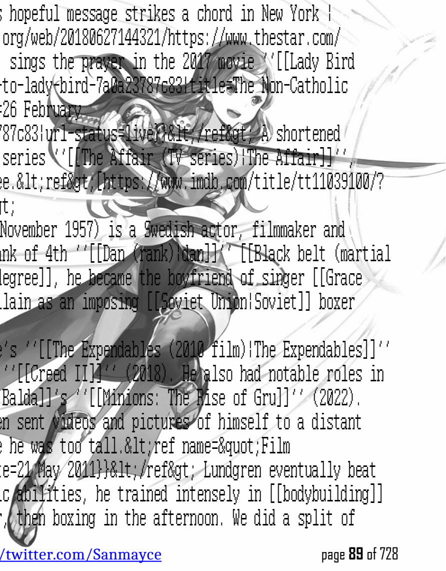
558 The advertisements were filmed in [[Los Angeles]], [[California]], featuring [[Sylvester Stallone]] and [[Gisele Bündchen]].&lt;ref&gt;{{cite web |url=http://www.germancarblog.com/2008/07/vw-gol-gisele-bundchen-and-sylvester.html |title=VW Gol: Gisele Bundchen and Sylvester Stallone starring in new commercial |date=2 July 2008 |access-date=23 October 2008 }}&lt;/ref&gt;

559 The prayer is referenced in the [[Alcoholics Anonymous]] book ''[[Twelve Steps and Twelve Traditions]]'' (1953), and is often known to AA members as the &quot;Step Eleven Prayer&quot;.&lt;ref&gt;{{cite book |title=Twelve Steps and Twelve Traditions |first=Bill |last=Wilson |year=1953 |page=99 |publisher=Alcoholics Anonymous |url=http://www.aa.org/assets/en_US/en_step11.pdf |isbn=978-0-916856-01-4 |access-date=23 March 2017 |archive-date=12 July 2017 |archive-url=https://web.archive.org/web/20170712162100/http://www.aa.org/assets/en_US/en_step11.pdf |url-status=live }}&lt;/ref&gt; An abbreviated version of the prayer was sung in [[Franco Zeffirelli]]'s 1972 film about St. Francis, ''[[Brother Sun, Sister Moon]]''.&lt;ref name=Donovan/&gt; In ''[[Band of Brothers (miniseries)|Band Of Brothers]]'' (2001), episode six &quot;Bastogne&quot;, Eugene 'Doc' Roe recites &quot;Lord, grant that I shall never seek so much to be consoled as to console, to be understood as to understand, or to be loved as to love with all my heart. With all my heart.&quot; while praying in a foxhole in the [[Bois Jacques]]. A modified segment of the prayer is recited in one of the early trailers for the [[Sylvester Stallone]] 2008 film ''[[Rambo (2008 film)|Rambo]]''.&lt;ref&gt;{{Citation|last=myxmovie|title=Rambo 4 (Trailer 2008)|date=15 January 2008|url=https://www.youtube.com/watch?v=DFtQ2TjfKo0|access-date=3 September 2018|archive-date=4 November 2018|archive-url=https://web.archive.org/web/20181104174951/https://www.youtube.com/watch?v=DFtQ2TjfKo0|url-status=live}}&lt;/ref&gt; A modified version of the prayer appears in the song &quot;Prayer&quot; in the musical ''[[Come From Away]]''.&lt;ref&gt;{{cite web |url=https://www.thestar.com/entertainment/stage/2017/03/05/come-from-away-has-an-urgent-message-will-america-listen.html |title=Come From Away's hopeful message strikes a chord in New York |last=Whyte |first=Murray |work=The Star |date=5 March 2017 |access-date=26 June 2018 |archive-date=27 June 2018 |archive-url=https://web.archive.org/web/20180627144321/https://www.thestar.com/entertainment/stage/2017/03/05/come-from-away-has-an-urgent-message-will-america-listen.html |url-status=live }}&lt;/ref&gt; [[Beanie Feldstein]] sings the prayer in the 2017 movie ''[[Lady Bird (film)|Lady Bird]]'', set at a Catholic girls' school.&lt;ref&gt;{{Cite web|url=https://medium.com/@johndoc86/the-non-catholic-school-kids-guide-to-lady-bird-7a0a23787c83|title=The Non-Catholic School Kid's Guide to Lady Bird - John Dougherty|last=Dougherty|first=John|date=13 June 2018|website=Medium|access-date=9 July 2019|archive-date=26 February 2019|archive-url=https://web.archive.org/web/20190226050020/https://medium.com/@johndoc86/the-non-catholic-school-kids-guide-to-lady-bird-7a0a23787c83|url-status=live}}&lt;/ref&gt; A shortened version appears in the [[HBO]] show ''[[Deadwood (TV series)|Deadwood]]'', episode 11, season one, and in the [[Showtime (TV network)|Showtime]] series ''[[The Affair (TV series)|The Affair]]'', episode 8, season one. And also appears in the [[Canadian Broadcasting Corporation|CBC]] TV series ''[[Anne with an E]]'', episode 3, season three.&lt;ref&gt;[https://www.imdb.com/title/tt11039100/?ref_=ttep_ep3 &quot;Anne with an E&quot; What Can Stop the Determined Heart (TV Episode 2019) - IMDb&lt;!-- Bot generated title --&gt;]&lt;/ref&gt;

560 '''Hans''' &quot;'''Dolph'''&quot; '''Lundgren''' ({{IPAc-en|'|l|ʌ|n|d|g|r|ən}}, {{IPA|sv|'dɔlːf 'lě̌nːdɡreːn|lang|Sv-Dolph_Lundgren.ogg}}; born 3 November 1957) is a Swedish actor, filmmaker and [[martial artist]]. Born in [[Spånga]], [[Sweden]], Lundgren became interested in martial arts at a young age. This would lead him to hold the rank of 4th ''[[Dan (rank)|dan]]'' [[Black belt (martial arts)|black belt]] in [[Kyokushin|Kyokushin karate]] and become European champion in 1980 and 1981. In 1982, while studying to get a [[master's degree]], he became the boyfriend of singer [[Grace Jones]]. He moved to [[New York City]] with her and started taking acting classes. In 1985, Lundgren had a breakthrough role playing the lead villain as an imposing [[Soviet Union|Soviet]] boxer named [[Ivan Drago]] in [[Sylvester Stallone]]'s ''[[Rocky IV]]''.

561 Lundgren returned to prominence in 2010 with the role of [[List of The Expendables characters#Gunner Jensen|Gunner Jensen]] in Sylvester Stallone's ''[[The Expendables (2010 film)|The Expendables]]'' alongside an all-action star cast. He reprised his role in its [[The Expendables (franchise)|sequels]]. He returned to the role of Ivan Drago in ''[[Creed II]]'' (2018). He also had notable roles in the [[Arrow season 5|fifth season]] of ''[[Arrow (TV series)|Arrow]]'' (2017), [[James Wan]]'s ''[[Aquaman (film)|Aquaman]]'' (2018), and [[Kyle Balda]]'s ''[[Minions: The Rise of Gru]]'' (2022).

562 Upon learning that [[Sylvester Stallone]] was seeking an imposing fighter to play Soviet boxer [[Ivan Drago]] in ''[[Rocky IV]]'' (1985), Lundgren sent videos and pictures of himself to a distant contact of Stallone, eventually reaching him. Lundgren tried out for the role, but as he himself has stated, he was initially turned down because he was too tall.&lt;ref name=&quot;Film review&quot;&gt;{{cite book |url=https://books.google.com/books?id=-ZcqAQAAIAAJ |title=Film review |publisher=Orpheus Pub. |year=1990 |access-date=21 May 2011}}&lt;/ref&gt; Lundgren eventually beat 5,000 other hopefuls to land his breakout role opposite Stallone, [[Carl Weathers]] and [[Brigitte Nielsen]]. To improve his physique and athletic abilities, he trained intensely in [[bodybuilding]] and [[boxing]] for five months before the film was shot. Lundgren said: &quot;We trained six days a week-weights in the morning for about an hour, then boxing in the afternoon. We did a split of



chest and back one day and then shoulders, legs, and arms the next. We boxed for an hour and a half, practiced the fight choreography, and did bag work and abs.&quot;&lt;ref name=&quot;FA&quot;&gt;{{cite magazine|url=http://findarticles.com/p/articles/mi_m0801/is_6_72/ai_n57642640/?tag=mantle_skin|title=He must break you: Dolph Lundgren reflects on playing Drago, crazed fans, and why you shouldn't break into his house|magazine=[[Muscle &amp; Fitness]]|access-date=4 October 2011|year=2011|archive-date=7 January 2016|archive-url=https://web.archive.org/web/20160107062450/http://findarticles.com/p/articles/mi_m0801/is_6_72/ai_n57642640/?tag=mantle_skin|url-status=dead}}&lt;/ref&gt; He weighed {{convert|243|lb|kg}} - {{convert|245|lb|kg}} during filming,&lt;ref name=&quot;FA&quot;/&gt; but in the film he was billed at {{convert|261|lb|kg}}; one publisher said of Drago, &quot;He's a hulking 261&amp;nbsp;pounds of merciless fighting machine, the best that Soviet science &amp; medicine can create&quot;.&lt;ref name=&quot;Bowker's Complete Video Directory 2002: Entertainment: titles A-S.&quot;&gt;{{cite book|title=Bowker's Complete Video Directory 2002: Entertainment : titles A-S.|url=https://books.google.com/books?id=KBJlAAAAMAAJ |access-date=22 May 2011|year=2002|publisher=[[R.R. Bowker]]|location=Chatham, New Jersey|isbn=978-0-8352-4478-7|page=1213}}&lt;/ref&gt; His character's lines &quot;If he dies, he dies&quot; and &quot;I must break you&quot; are amongst the best known of the [[Rocky (franchise)|''Rocky'' series]], and have often been cited in popular culture.&lt;ref name=&quot;Kluck2009&quot;&gt;{{cite book|last=Kluck|first=Ted|title=The Reason for Sports: A Christian Fanifesto |url=https://books.google.com/books?id=i-I3IR838XgC&amp;pg=PA117|access-date=22 May 2011|date=1 August 2009|publisher=Moody Publishers|location=Chicago, Illinois|isbn=978-0-8024-5836-0|page=117}}&lt;/ref&gt;&lt;ref name=&quot;Allred2009&quot;&gt;{{cite book|last=Allred|first=Lance|title=Longshot: The Adventures of a Deaf Fundamentalist Mormon Kid and His Journey to the NBA|url=https://books.google.com/books?id=pnENAQAAMAAJ|access-date=22 May 2011|date=15 May 2009|publisher=[[HarperCollins|HarperOne]]|location=New York City|isbn=978-0-06-171858-8}}&lt;/ref&gt; Lundgren hit Stallone so hard during the filming of a fight scene that Stallone was in intensive care in the St. John's Hospital for nine days with a blood pressure of 290, due to swelling of the [[pericardial sac]] around his heart.&lt;ref&gt;{{cite magazine|first=Mandi |last=Bierly|date=22 December 2006|title=Why 'Rocky IV' is the best fight-Ever!|magazine=[[Entertainment Weekly]]|url=https://ew.com/article/2006/12/22/why_rocky_iv_is/|quote=This accident is not in keeping with boxing. Usually this kind of pericardial swelling is the result of head on collisions, when the steering wheel hits you in the chest.|access-date=7 November 2009}}&lt;/ref&gt; Lundgren later fought in a real boxing match against former [[UFC]] fighter [[Oleg Taktarov]], and lost via decision.&lt;ref&gt;Strength Fighter, 5 June 2013. Video of [https://www.strengthfighter.com/2013/06/dolph-lundgren-vs-oleg-taktarov-boxing.html Dolph Lundgren and Oleg Taktarov Boxing Match] {{Webarchive|url=https://web.archive.org/web/20210416223418/https://www.strengthfighter.com/2013/06/dolph-lundgren-vs-oleg-taktarov-boxing.html |date=16 April 2021 }}. {{Retrieved|access-date=4 March 2021}}&lt;/ref&gt; Lundgren has highlighted the premiere of ''Rocky IV'' at the [[Fox Theater, Westwood Village|Mann Village Theatre]],&lt;ref&gt;{{cite web | url=https://www.gettyimages.com/detail/news-photo/the-marquee-of-the-fox-westwood-village-theatre-heralds-the-news-photo/79796220 | title=The marquee of the Fox Westwood Village Theatre heralds the 1985 West | date=16 February 2008 }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://www.gettyimages.com.au/detail/news-photo/american-actor-tony-curtis-and-his-wife-andrea-savio-with-news-photo/1391055255 | title=American actor Tony Curtis and his wife, Andrea Savio, with Canadian | date=12 April 2022 }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://www.gettyimages.com/detail/news-photo/american-actress-lydia-cornell-wearing-a-red-outfit-and-news-photo/1849766087 | title=American actress Lydia Cornell, wearing a red outfit, and American | date=14 December 2023 }}&lt;/ref&gt; in [[Westwood, Los Angeles]], as the moment which changed his life, remarking: &quot;I walked in to a Westwood movie theater as Grace Jones' boyfriend and walked out ninety minutes later as the movie star Dolph Lundgren. I was shell-shocked for years from the mind-boggling and daunting experience of being a student-athlete from tiny Sweden suddenly having to live up a new action-star persona.&quot;&lt;ref name=&quot;The Official biography&quot;/&gt;

563 Although Lundgren has never competed as a professional bodybuilder, he has been closely associated with bodybuilding and fitness since his role as Drago in the mid-1980s. [[Bodybuilding.com]] said, &quot;Looking like a man in his 30s rather than his 50s, Lundgren is the poster boy of precise nutrition, supplementation and exercise application that he has practiced for over 35 years&quot;.&lt;ref name=&quot;BBI&quot;&gt;{{citation|title=Dolph Lundgren Interview|publisher=Bodybuilding.com}}&lt;/ref&gt; In an interview with them, he claimed to often train up to six days a week, usually one-hour sessions completed in the morning, saying that &quot;it's just one hour a day, and then you can enjoy the other 23 hours&quot;.&lt;ref name=&quot;Bodybuilding&quot;&gt;{{cite web|url=https://www.youtube.com/watch?v=oboq-208Xx8 | archive-url=https://ghostarchive.org/varchive/youtube/20211030/oboq-208Xx8| archive-date=30 October 2021|title=Dolph Lundgren: Interview (Part 1)|publisher=Youtube.com|access-date=22 May 2011}}{{cbignore}}&lt;/ref&gt; Although he had begun lifting weights as a teenager, he cites co-star [[Sylvester Stallone]] as the man who got him into serious bodybuilding for a period in the 1980s after he arrived in the U.S.&lt;ref name=&quot;Bodybuilding&quot;/&gt; Stallone had a lasting influence on his fitness regimen and diet, ensuring that he ate a much higher percentage of protein and split his food intake between five or six smaller meals a day.&lt;ref name=&quot;BBI&quot;/&gt; In 2023, he said in an interview that he took anabolic steroids off and on from the mid-'80s to the mid-'90s.&lt;ref&gt;{{Cite web |last= |first= |date= 11 May 2023|title=Dolph Lundgren Admits to Taking Steroids |url=https://thebarbell.com/dolph-lundgren-steroids/ |access-date=2023-11-10 |website=The Barbell |language=en-US}}&lt;/ref&gt; Lundgren has professed never to have been &quot;super strong&quot;, saying that, &quot;I'm too tall and my arms are long. I think back then [''Rocky IV''] I was working with around 300 pounds on the bench and squat.&quot;&lt;ref name=&quot;FA&quot;/&gt;

564 Everhart was married to [[Ashley Hamilton]] from December 1, 1996, until their divorce in March 1997. [[Sylvester Stallone]] and Everhart were briefly engaged in 1995, but they never married. &lt;!--briefly dated [[Howard Stern]] in late 2000 and--&gt;She was &lt;!--briefly--&gt; engaged to &lt;!-- [[Sylvester Stallone]]and Angie Everhart were briefly engaged in 1995, but they never made it down the aisle. in 1995 and--&gt; [[Joe Pesci]],&lt;!--in 2007.--&gt;&lt;ref name=&quot;sfgate&quot;&gt;{{cite news | url = http://www.sfgate.com/cgi-bin/blogs/sfgate/detail?blogid=7&amp;entry_id=19023 | title=Pesci to Wed Everhart | access-date=July 31, 2007 | date=July 30, 2007 | work =[[San Francisco Chronicle]] | first=Karen | last=Reardanz}}&lt;/ref&gt; but the couple broke up in 2008.&lt;ref&gt;{{cite web|url= http://breakingnews.iol.ie/entertainment/story.asp?j=253669330&amp;p=z5367xx36|title= Pesci ends engagement with model Everhart|access-date= April 25, 2008|date= April 25, 2008|publisher= Ireland On-line|archive-url= https://web.archive.org/web/20080430170839/http://breakingnews.iol.ie/entertainment/story.asp?j=253669330&amp;p=z5367xx36|archive-date= April 30, 2008|url-status= dead}}&lt;/ref&gt;

565 After her breakthrough in ''Northern Exposure'', Turner appeared in the action film ''[[Cliffhanger (film)|Cliffhanger]]'' opposite [[Sylvester Stallone]]. She next appeared as [[June Cleaver]] in a [[Leave It to Beaver (film)|''Leave It to Beaver'' film adaptation]] of television's original ''[[Leave it to Beaver]]'', then in ''[[Stolen Women: Captured Hearts]]'', and ''[[Dr. T &amp; the Women]]'' with [[Richard Gere]]. She also appeared in ''No Regrets'' and numerous movies of the week.

566 Turner has never married. Her daughter, Juliette Turner-Jones, was born in 1997.&lt;ref&gt;{{cite web | title=Names and Faces: Janine Turner Trades Northwest For Southern Exposure On Ranch | date=July 1, 1997 | url=https://www.orlandosentinel.com/1997/07/01/janine-turner-trades-northwest-for-southern-exposure-on-ranch/ | publisher=[[Orlando Sentinel]] | access-date=October 25, 2015}}&lt;/ref&gt; Turner was once engaged to [[Alec Baldwin]] and also dated [[Troy Aikman]], [[Mikhail Baryshnikov]], [[Mark Grace]] and [[Sylvester Stallone]].&lt;ref&gt;{{cite web | url=http://www.people.com/people/archive/article/0,,20150702,00.html | title=Janine Turner | date=August 2, 2004 | work=[[People (magazine)|People]] | access-date=December 2, 2016}}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=http://movieline.com/1997/07/01/janine-turners-life-story/ | title=Janine Turner's Life Story | first=Gabriel | last=Koen | date=July 2, 1997 | work=[[Movieline]] | access-date=December 2, 2016}}&lt;/ref&gt; As of 2014, Turner was living on a [[Texas longhorn (cattle)|longhorn cattle]] ranch outside [[Dallas]] with her daughter.&lt;ref name=&quot;Council&quot;&gt;{{cite web |url=http://www.usafreedomcorps.gov/council/members/index.asp |title=Meet the Council Members | archive-url=https://web.archive.org/web/20080411040001/http://www.usafreedomcorps.gov/council/members/index.asp |archive-date=April 11, 2008 |date=2006| website=USA Freedom Corps | publisher=The White House, [[President's Council on Service and Civic Participation]]|quote= Ms. Turner lives on her longhorn cattle ranch in Texas with her eight-year-old daughter, Juliette Turner-Jones }}&lt;/ref&gt;

567 [[Harrison Ford]] was offered the role of Sam Daniels, but turned it down.&lt;ref&gt;{{Cite news|url=http://www.nytimes.com/1995/03/19/movies/film-wrestling-a-virus-to-the-screen.html?pagewanted=all|title = FILM; Wrestling a Virus to the Screen|newspaper = The New York Times|date = March 19, 1995|last1 = Weinraub|first1 = Bernard}}&lt;/ref&gt; [[Sylvester Stallone]] and [[Mel Gibson]] were also offered the role, but turned it down.&lt;ref&gt;{{Cite web|url=https://www.eonline.com/news/1129426/25-shocking-secrets-about-outbreak-a-surprising-real-housewives-connection-and-more|title = 25 Shocking Secrets About Outbreak - E! Online| date=March 10, 2020 }}&lt;/ref&gt;

568 * [[Sylvester Stallone]]

569 * [[Sylvester Stallone]]

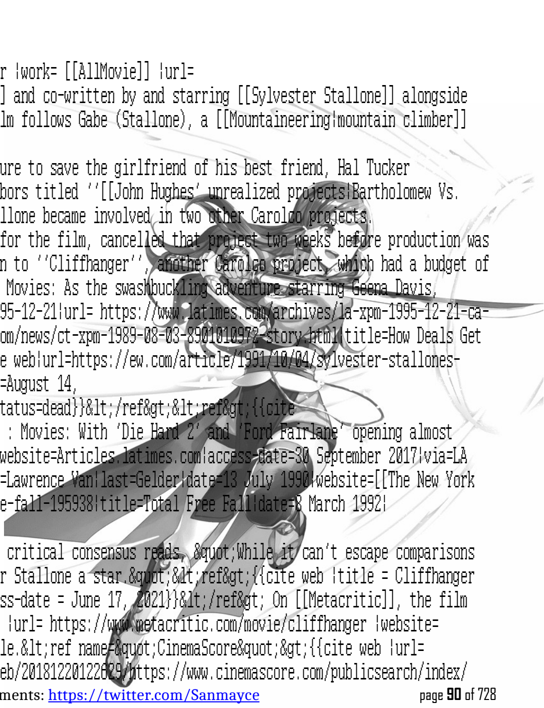
570 '''''Cliffhanger''''' is a 1993 American [[action thriller film]]&lt;ref name=&quot;AllMovie&quot;&gt;{{cite web |title= Cliffhanger |work= [[AllMovie]] |url= https://www.allmovie.com/movie/cliffhanger-v9983 |access-date= 2021-09-02}}&lt;/ref&gt; directed and co-produced by [[Renny Harlin]] and co-written by and starring [[Sylvester Stallone]] alongside [[John Lithgow]], [[Michael Rooker]] and [[Janine Turner]]. Based on a concept by climber [[John Long (climber)|John Long]], the film follows Gabe (Stallone), a [[Mountaineering|mountain climber]] who becomes embroiled in a [[Robbery|heist]] of a U.S. Treasury plane flying through the [[Rocky Mountains]].

571 * [[Sylvester Stallone]] as Ranger Gabriel &quot;Gabe&quot; Walker, a former mountain climber and rescue ranger haunted by his failure to save the girlfriend of his best friend, Hal Tucker

572 [[Carolco Pictures]] had originally signed [[Sylvester Stallone]] to appear opposite [[John Candy]] in a comedy about feuding neighbors titled ''[[John Hughes' unrealized projects|Bartholomew Vs. Neff]]'', which was going to be written and directed by [[John Hughes (filmmaker)|John Hughes]]. When that project was dropped, Stallone became involved in two other Carolco projects.

573 Carolco, believing the intended $40&amp;nbsp;million budget would be too big, and unable to figure out how to make special effects for the film, cancelled that project two weeks before production was supposed to begin; but Harlin still kept his $3&amp;nbsp;million, and he and Stallone and everyone else involved in it then moved on to ''Cliffhanger'', another Carolco project, which had a budget of $70&amp;nbsp;million, almost double that of ''Gale Force''.&lt;ref&gt;{{cite news|title= Troubled Route to Pirate Epic 'Cutthroat'; Movies: As the swashbuckling adventure starring Geena Davis, directed by her husband, Renny Harlin, opens this weekend, financial woes surround its release.|work= The Los Angeles Times|date=1995-12-21|url= https://www.latimes.com/archives/la-xpm-1995-12-21-ca-16222-story.html|access-date=2010-10-27|first=Judy|last=Brennan}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.chicagotribune.com/news/ct-xpm-1989-08-03-8901010972-story.html|title=How Deals Get Done Over A Hot Script|website=Article.latimes.com|date=August 3, 1989 |access-date=30 September 2017}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://ew.com/article/1991/10/04/sylvester-stallones-gale-force/|title=Sylvester Stallone's ''Gale Force''|date=4 October 1991|website=Ew.com|access-date=30 September 2017|archive-date=August 14, 2020|archive-url=https://web.archive.org/web/20200814115817/https://ew.com/article/1991/10/04/sylvester-stallones-gale-force/|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.latimes.com/archives/la-xpm-1990-07-04-ca-273-story.html|title=Renny Harlin Finds Plenty of Action in Hollywood : Movies: With 'Die Hard 2' and 'Ford Fairlane' opening almost simultaneously, the Finnish director of adventure films is taking the industry by storm.|first=CHRIS|last=WILLMAN|date=4 July 1990|website=Articles.latimes.com|access-date=30 September 2017|via=LA Times}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.nytimes.com/1990/07/13/movies/at-the-movies.html|title=AT THE MOVIES|first=Lawrence Van|last=Gelder|date=13 July 1990|website=[[The New York Times]]|access-date=30 September 2017|via=www.nytimes.com}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.newsweek.com/total-free-fall-195938|title=Total Free Fall|date=8 March 1992|website=Europe.newsweek.com|access-date=1 October 2017}}&lt;/ref&gt;

574 On [[Rotten Tomatoes]], the film has an approval rating of 68% based on 55 reviews, with an average rating of 6.2/10. The website's critical consensus reads, &quot;While it can't escape comparisons to the movies it borrows from, ''Cliffhanger'' is a tense, action-packed thriller and a showcase for the talents that made Sylvester Stallone a star.&quot;&lt;ref&gt;{{cite web |title = Cliffhanger (1993) |url = https://www.rottentomatoes.com/m/1044214-cliffhanger/ |work = [[Rotten Tomatoes]] |publisher=[[Fandango Media]] |access-date = June 17, 2021}}&lt;/ref&gt; On [[Metacritic]], the film has a score of 60 out of 100 based on reviews from 16 critics.&lt;ref name=&quot;metacritic&quot;&gt;{{cite web |title= Cliffhanger |url= https://www.metacritic.com/movie/cliffhanger |website= [[Metacritic]] }}&lt;/ref&gt; Audiences polled by [[CinemaScore]] gave the film an average grade of &quot;B&quot; on an A+ to F scale.&lt;ref name=&quot;CinemaScore&quot;&gt;{{cite web |url= https://www.cinemascore.com/publicsearch/index/title/ |title= Cinemascore |url-status= dead |archive-url= https://web.archive.org/web/20181220122629/https://www.cinemascore.com/publicsearch/index/



title/ |archive-date= 2018-12-20 }}&lt;/ref&gt;

575 In 1994, TriStar announced plans to develop a sequel titled ''Cliffhanger 2: The Dam'', with Stallone reprising his starring role. The plot revolved around Gabe Walker combating terrorists who took control of the [[Hoover Dam]]. The project remained in [[development hell]] until 2008, when the project was revived with Stallone's involvement, before once again being shelved.&lt;ref name=&quot;Sequel_SF&quot;&gt;{{cite web|url=https://www.slashfilm.com/sylvester-stallone-to-star-in-cliffhanger-2-the-dam/|title=Sylvester Stallone to Star in Cliffhanger 2: The Dam? - /Film|date=16 February 2008|website=Slashfilm.com|access-date=30 September 2017}}&lt;/ref&gt; By May 2009, it was announced that [[Reboot (fiction)|reimagining]] was in development. Produced by [[Neal H. Moritz]] the project would be a joint-production between [[Original Film]] and [[StudioCanal]]. The story would focus around a group of young climbers, and was tentatively scheduled to begin principal photography the following year. Moritz stated that his intent is to adapt the story in a similar manner comparable to [[J.J. Abrams]]' work on [[List of Star Trek films#Reboot (Kelvin timeline) films|''Star Trek'']].&lt;ref name=&quot;Remake_Variety&quot;&gt;{{cite magazine|url= https://www.variety.com/article/VR1118003577 |title=StudioCanal remounts 'Cliff' - Entertainment News, Cannes News, Media |magazine=Variety |date= May 13, 2009|access-date=May 14, 2009}}&lt;/ref&gt; By May 2014, Joe Gazzam was hired as screenwriter after [[Pitch (filmmaking)|pitching]] his approach to the story to Moritz who was impressed.&lt;ref name=&quot;Remake_Deadline&quot;&gt;{{cite news|title=Climb Toward 'Cliffhanger' Reboot Moving Forward; Joe Gazzam Set To Write|url=https://www.deadline.com/2014/05/climb-toward-cliffhanger-reboot-moving-forward-joe-gazzam-set-to-write/|access-date=30 May 2014|newspaper=deadline.com}}&lt;/ref&gt; By May 2015, Stallone expressed interest in developing a direct sequel to the original film.&lt;ref name=&quot;Sequel_Stallone&quot;&gt;{{cite web |date=May 31, 2015 |title=Sylvester Stallone teases 'Cliffhanger' sequel (Photo) |url=https://fansided.com/2015/05/31/sylvester-stallone-cliffhanger-sequel-photo/ |access-date=May 1, 2018 |publisher=FanSided |archive-date=June 1, 2015 |archive-url=https://web.archive.org/web/20150601075257/https://fansided.com/2015/05/31/sylvester-stallone-cliffhanger-sequel-photo/ |url-status=dead }}&lt;/ref&gt; In May 2019, the project developed into a female-led adaptation. [[Ana Lily Amirpour]] was hired as director, with a new draft of the script written by Sascha Penn. [[Jason Momoa]] was in early negotiations to feature in a prominent cameo role; while Mortiz brought on Toby Jaffe, Thorsten Schumacher, and Lars Sylvest as additional producers. The story was described as a survival thriller action movie, with elements of espionage. An official production poster was released for the project's presence at Cannes, with a tentative commencement for principal photography was set for 2020. Moritz stated that there are plans for more than one installment to be made.&lt;ref name=&quot;Remake_Deadline2&quot;&gt;{{cite web |url=https://deadline.com/2019/05/cliffhanger-reboot-jason-momoa-ana-lily-amirpour-cannes-female-1202608726/ |title=Hang On, Cannes! Neal Moritz's Female-Fronted 'Cliffhanger' reboot Climbs With Rocket Science, CAA |last=Wiseman |first=Andreas |website=[[Deadline Hollywood]] |date=May 8, 2019 |access-date=May 8, 2019 |archive-url=https://web.archive.org/web/20190508222223/https://deadline.com/2019/05/cliffhanger-reboot-jason-momoa-ana-lily-amirpour-cannes-female-1202608726/ |archive-date=May 8, 2019}}&lt;/ref&gt;

576 In May 2023, it was officially announced that the project will be redeveloped as a legacy-sequel. [[Ric Roman Waugh]] will serve as director (replacing Amirpour), from a new script written by Mark Bianculli. Stallone will reprise his role from the original, in addition to taking on a role as producer. The plot will detail the continued adventures of climber-turned-rescue ranger Gabriel &quot;Gabe&quot; Walker alongside a supporting cast, and include the [[Alps#Geology and orogeny|Italian Alps]]. Casting underway for additional ensemble lead and supporting roles. Waugh expressed excitement for the challenge of working with Stallone, stating: &quot;Growing up with the biggest action films of the '80s and '90s, ...''Cliffhanger'' was by far one of my favorite spectacles. To be at the helm of the next chapter, ...with the legend himself...is a dream come true. It's going to be a great challenge and blast taking this franchise to new heights, a responsibility I don't take lightly.&quot; Moritz, Jaffe, Schumacher, Sylvest, and Braden Aftergood will serve as additional producers. The project will be a joint-venture production between Original Film, [[Balboa Productions]], StudioCanal, Rocket Science Films, Wright Productions &amp; Entertainment, and Front Row Entertainment. The movie had a presence at Cannes, where additional funding and distribution would be decided.&lt;ref name=&quot;Sequel_Deadline&quot;&gt;{{cite web|url=https://deadline.com/2023/05/sylvester-stallone-set-for-cliffhanger-reboot-from-director-ric-roman-waugh-1235351858/|work=Deadline|title=Sylvester Stallone Set For 'Cliffhanger' Reboot From Director Ric Roman Waugh|author=Grobar, Matt|date=May 1, 2023|accessdate=May 4, 2023}}&lt;/ref&gt;&lt;ref name=&quot;Sequel_THR&quot;&gt;{{cite web|url=https://www.hollywoodreporter.com/movies/movie-news/cliffhanger-reboot-set-sylvester-stallone-1235476408/|work=The Hollywood Reporter|title=Sylvester Stallone Sets 'Cliffhanger' Reboot With 'Angel Has Fallen' Director|author=Kit, Borys|date=May 1, 2023|accessdate=May 4, 2023}}&lt;/ref&gt;&lt;ref name=&quot;Sequel_Variety&quot;&gt;{{cite web|url=https://variety.com/2023/film/news/sylvester-stallone-cliffhanger-reboot-ric-roman-waugh-1235599620/|work=Variety|title=Sylvester Stallone Returning for 'Cliffhanger' Reboot, Ric Roman Waugh Directing|author=Lang, Brent|date=May 1, 2023|accessdate=May 4, 2023}}&lt;/ref&gt; In December of the same year, it was announced that [[Jean-François Richet]] had replaced Waugh as director; while additional film studios joined the project including [[FilmFernsehFonds Bayern]], Black Magic Films, Supernix, Maze Pictures, and Occupant Entertainment. Philipp Kreuzer and Joe Neurauter will serve as additional producers, while principal photography is scheduled to commence in the summer of 2024.&lt;ref name=&quot;Sequel_SD&quot;&gt;{{cite web|url=https://www.screendaily.com/news/sylvester-stallones-cliffhanger-sequel-wins-backing-from-german-fund-ahead-of-summer-2024-shoot/5188846.article|work=Screen Daily|title=Sylvester Stallone's 'Cliffhanger' sequel wins backing from German fund ahead of summer 2024 shoot|author=Blaney, Martin|date=December 15, 2023|accessdate=June 11, 2024}}&lt;/ref&gt;

577 {{Sylvester Stallone}}

578 [[Category:Films with screenplays by Sylvester Stallone]]

579 The Cannon Group's biggest financial success was with the 1986 action film ''[[Cobra (1986 film)|Cobra]]'', which starred [[Sylvester Stallone]]; not a low-budget film, it earned $160 million on a $25 million budget.

580 In 1987, IVE was acquired by [[Carolco Pictures]] from NCB Entertainment after Carolco had taken a minority interest in the latter a year earlier.&lt;ref name=&quot;More&quot;&gt;{{Cite web |title=Artisan Entertainment Inc. - Company Profile on Artisan Entertainment Inc |url=http://www.referenceforbusiness.com/history2/20/Artisan-Entertainment-Inc.html |url-status=live |archive-url=https://web.archive.org/web/20110615114133/http://www.referenceforbusiness.com/history2/20/Artisan-Entertainment-Inc.html |archive-date=June 15, 2011 |access-date=July 12, 2011 |publisher=Referenceforbusiness.com}}&lt;/ref&gt;&lt;ref&gt;{{Cite book |url=https://books.google.com/books?id=diQEAAAAMBAJ&amp;pg=PA1 |title=Billboard |date=February 8, 1986 |page=1 |issn=0006-2510}}&lt;/ref&gt;&lt;ref&gt;{{Cite book |url=https://books.google.com/books?id=VyQEAAAAMBAJ&amp;pg=PA6 |title=Billboard |date=June 28, 1986 |page=6 |issn=0006-2510}}&lt;/ref&gt;&lt;ref name=&quot;google3&quot;&gt;{{Cite book |last1=Reed, R.M. |url=https://books.google.com/books?id=gU3uBwAAQBAJ&amp;pg=PA320 |title=The Encyclopedia of Television, Cable, and Video |last2=Reed, M.K. |date=2012 |publisher=Springer US |isbn=9781468465211 |page=320}}&lt;/ref&gt;&lt;ref&gt;{{Cite book |url=https://books.google.com/books?id=qyQEAAAAMBAJ&amp;pg=PT69 |title=Billboard |date=July 19, 1986 |pages=66 |issn=0006-2510}}&lt;/ref&gt; That year, it secured a deal with MCA Distributing Corporation to handle distribution of its titles.&lt;ref&gt;{{Cite magazine |last=McCallaugh |first=Jim |date=February 28, 1987 |title=IVE Is Optimistic About Move To MCA Distribution |url=https://www.worldradiohistory.com/hd2/IDX-Business/Music/Billboard-Index/IDX/1987/1987-02-28-Billboard-Page-0006.pdf |access-date=April 7, 2024 |magazine=[[Billboard (magazine)|Billboard]] |pages=6, 77}}&lt;/ref&gt; The unrated release of ''[[Angel Heart]]'' was the first Carolco film released by IVE on video. The studio hired José Menendez, previously of [[RCA]], as head of IVE; he was responsible for creating product deals with [[Sylvester Stallone]]'s White Eagle Enterprises and producer [[Edward Pressman]].&lt;ref name=&quot;More&quot; /&gt; In 1989, Menendez and his wife were murdered by [[Lyle and Erik Menendez|their two sons]].&lt;ref name=&quot;More&quot; /&gt;&lt;ref&gt;{{Cite news |date=August 22, 1989 |title=Video Company Chief, Wife Found Fatally Shot in Mansion Home |work=Associated Press |url=https://apnews.com/article/7aaededea59a41e38b771e66187fd019 |url-status=live |access-date=July 7, 2021 |archive-url=https://web.archive.org/web/20210709183049/https://apnews.com/article/7aaededea59a41e38b771e66187fd019 |archive-date=July 9, 2021}}&lt;/ref&gt; Also in 1987, Noel C. Bloom left IVE, following disputes with Carolco, to start out [[Celebrity Home Entertainment]], with some of IVE's employees defecting to Celebrity.&lt;ref&gt;{{Cite magazine |last=Stewart |first=Al |date=May 9, 1987 |title=Newsline |magazine=[[Billboard (magazine)|Billboard]] |url=https://worldradiohistory.com/hd2/IDX-Business/Music/Billboard-Index/IDX/1987/1987-05-09-Billboard-Page-0065.pdf |url-status=live |access-date=October 22, 2021 |archive-url=https://web.archive.org/web/20211022021910/https://worldradiohistory.com/hd2/IDX-Business/Music/Billboard-Index/IDX/1987/1987-05-09-Billboard-Page-0065.pdf |archive-date=October 22, 2021}}&lt;/ref&gt;&lt;ref&gt;{{Cite magazine |last=McCormick |first=Moira |date=June 6, 1987 |title=Celebrity Home Video |magazine=[[Billboard (magazine)|Billboard]] |url=https://worldradiohistory.com/hd2/IDX-Business/Music/Billboard-Index/IDX/1987/1987-06-06-Billboard-Page-0090.pdf |url-status=live |access-date=October 22, 2021 |archive-url=https://web.archive.org/web/20211022021911/https://worldradiohistory.com/hd2/IDX-Business/Music/Billboard-Index/IDX/1987/1987-06-06-Billboard-Page-0090.pdf |archive-date=October 22, 2021}}&lt;/ref&gt; Later that year, the company had acquired the assets of home video distributor Vista Home Video from The Vista Organization for $38 million.&lt;ref&gt;{{Cite news |last=Bierbaum |first=Tom |date=December 2, 1987 |title=IVE To Acquire All Vista Stock; On Heels of Heron Deal Dispute |page=90 |work=[[Variety (magazine)|Variety]] |url=https://varietyultimate.com/archive/issue/WV-12-02-1987-90 |url-status=live |access-date=April 17, 2022 |archive-url=https://web.archive.org/web/20220417052918/https://varietyultimate.com/archive/issue/WV-12-02-1987-90 |archive-date=April 17, 2022}}&lt;/ref&gt;

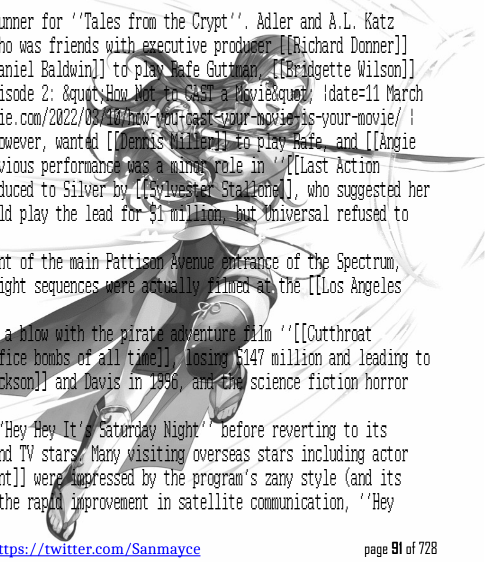
581 * [[Sylvester Stallone]]&lt;ref name=&quot;RS&quot; /&gt;

582 The production budget was $2.5 million.&lt;ref name=TaintedBlood/&gt; [[Gilbert Adler]] was hired to direct, having previously been a showrunner for ''Tales from the Crypt''. Adler and A.L. Katz rewrote Gale and Zemeckis' script to make the film more modern.&lt;ref name=TaintedBlood/&gt;&lt;ref name=Exclaim/&gt; [[Corey Feldman]], who was friends with executive producer [[Richard Donner]] and had previously acted in an episode of ''Tales'', was cast in the film as Caleb.&lt;ref name=TaintedBlood/&gt; Adler and Katz wanted [[Daniel Baldwin]] to play Rafe Guttman, [[Bridgette Wilson]] to play Katherine&lt;ref&gt;{{Cite web |url=https://how-not-to-make-a-movie.com/2022/03/10/how-you-cast-your-movie-is-your-movie/ |title=Episode 2: &quot;How Not to CAST a Movie&quot; |date=11 March 2022 |access-date=2023-11-17 |archive-date=2023-11-17 |archive-url=https://web.archive.org/web/20231117231942/https://how-not-to-make-a-movie.com/2022/03/10/how-you-cast-your-movie-is-your-movie/ |url-status=live }}&lt;/ref&gt; and [[Robin Givens]] to play [[Lilith]].&lt;ref name=TaintedBlood/&gt; Executive producer [[Joel Silver]], however, wanted [[Dennis Miller]] to play Rafe, and [[Angie Everhart]] to play Lilith. Silver had recently produced a movie with a [[supermodel]] in a leading role, and while Everhart, whose only previous performance was a minor role in ''[[Last Action Hero]]'', was not a trained actress, Silver thought &quot;supermodel actresses&quot; was &quot;the next big thing&quot;. Everhart was introduced to Silver by [[Sylvester Stallone]], who suggested her for the part of Lilith.&lt;ref name=TaintedBlood/&gt;&lt;ref name=TaintedBlood2/&gt; Miller, who did not want to make the film, said he would play the lead for $1 million, but Universal refused to put up this salary for Miller, so Silver cut $750,000 from the special effects budget to hire Miller.&lt;ref name=TaintedBlood/&gt;

583 A [[statue]] of [[Sylvester Stallone]], depicted in his role of Philadelphia [[boxing|boxer]] [[Rocky Balboa]], stood for many years in front of the main Pattison Avenue entrance of the Spectrum, which had been represented in the movie as the site of Rocky's [[Rocky|first]] and [[Rocky II|second]] fights with [[Apollo Creed]]. (The fight sequences were actually filmed at the [[Los Angeles Memorial Sports Arena]].) The statue was removed several times over the years to be used in the filming of sequels to the original film.

584 1993's action thriller ''[[Cliffhanger (film)|Cliffhanger]]'' was Harlin's first film with [[Sylvester Stallone]]. Harlin's career suffered a blow with the pirate adventure film ''[[Cutthroat Island]]'' in 1995, which starred Harlin's then-wife, [[Geena Davis]]. ''Cutthroat Island'' was one of the biggest [[box-office bomb|box-office bombs of all time]], losing $147 million and leading to the bankruptcy of [[Carolco Pictures]]. Harlin did go on to have moderate success with ''[[Long Kiss Goodnight]]'', starring [[Samuel L. Jackson]] and Davis in 1996, and the science fiction horror film ''[[Deep Blue Sea (1999 film)|Deep Blue Sea]]'' in 1999.

585 In 1984, the Nine Network moved the show from its morning timeslot to a [[primetime]] slot on Saturday nights, and it was briefly renamed ''Hey Hey It's Saturday Night'' before reverting to its original name. During this time, ''Hey Hey'' also became one of the most important TV venues for both local and international music, film and TV stars. Many visiting overseas stars including actor [[Sylvester Stallone]], singer [[Tom Jones (singer)|Tom Jones]], musician [[Stevie Ray Vaughan]], and professional wrestler [[André the Giant]] were impressed by the program's zany style (and its wide appeal) and made return appearances on subsequent visits. During its peak years, backed by the full resources of Nine and assisted by the rapid improvement in satellite communication, ''Hey Hey'' regularly travelled to locations all over Australia and even overseas for live broadcasts.



586 As an homage, actor [[Sylvester Stallone]] named his pet [[Bullmastiff]] ''Butkus'' after the dog ate a security blanket. He decided to name him after &quot;possibly the fiercest football player in history&quot;.&lt;ref&gt;{{cite book|last1=Shepatin|first1=Matthew|title=&quot;Then Madden Said to Summerall. . .&quot;: The Best NFL Stories Ever Told|date=2009|publisher=Triumph Books|isbn=978-1617491986|page=134|url=https://books.google.com/books?id=eBN5BweHdvUC&amp;pg=PT134|access-date=August 2, 2017}}&lt;/ref&gt; The dog later starred alongside Stallone in the ''[[Rocky (film series)|Rocky]]'' film series.&lt;ref name=Mullin/&gt;

587 In 2010, Statham appeared alongside fellow action stars [[Sylvester Stallone]], [[Jet Li]], [[Dolph Lundgren]] and [[Mickey Rourke]], among others, in the [[Ensemble cast|ensemble]] [[action film]] ''[[The Expendables (2010 film)|The Expendables]]''. Statham plays [[List of characters from The Expendables film series#Lee Christmas|Lee Christmas]], a former SAS soldier and expert at [[close quarters combat]] using knives.&lt;ref&gt;{{Cite news|url=https://www.vanityfair.com/hollywood/2017/04/arnold-schwarzenegger-expendables|title=Arnold Schwarzenegger Is Exiting The Expendables Because He Felt Too Expendable|last=Desta|first=Yohana|work=HWD|access-date=13 February 2018}}&lt;/ref&gt; The film was commercially successful, opening at number one at the box office in the United States, the United Kingdom, China and India, and grossed a total of $274 million worldwide.

588 In August 2011, he began filming ''[[Parker (2013 film)|Parker]]'' for director [[Taylor Hackford]]; the film was released in January 2013. Statham played [[Parker (fictional criminal)|Parker]], the criminal antihero previously played by [[Mel Gibson]] in 1999's ''[[Payback (1999 film)|Payback]]'' and by [[Lee Marvin]] in 1967's ''[[Point Blank (1967 film)|Point Blank]]'' (though their characters were given different surnames).&lt;ref&gt;{{cite web|url=https://www.deadline.com/2011/06/filmdistrict-acquires-taylor-hackford-helmed-parker-with-jason-statham/|title=FilmDistrict Acquires Taylor Hackford-Helmed 'Parker' With Jason Statham|author=Mike Fleming (writer)|date=20 June 2011|website=[[Deadline Hollywood]]|access-date=6 February 2014}}&lt;/ref&gt; [[A. O. Scott]] of ''The New York Times'' said of the actor in the film: &quot;[Statham], who seems to be made entirely of muscle and scar tissue, is comfortable with his limitations as an actor. His Parker, in any case, is more of an axiom than a fully rounded human being.&quot;&lt;ref&gt;{{Cite news|url=https://www.nytimes.com/2013/01/25/movies/parker-starring-jennifer-lopez-and-jason-statham.html|title='Parker,' Starring Jennifer Lopez and Jason Statham|last=Scott|first=A. O.|date=2013|work=The New York Times|access-date=15 January 2018|issn=0362-4331}}&lt;/ref&gt; A 2012 [[BBC News]] report estimated that his ten-year film career to date (2002 to 2012) yielded over one billion dollars in the box office, making him one of the industry's most [[Bankable star|bankable]] stars.&lt;ref name=&quot;:2&quot; /&gt; He was signed on to reprise his role as Lee Christmas in ''[[The Expendables 2]]'' in 2012.&lt;ref&gt;{{Cite news|url=https://deadline.com/2018/01/sylvester-stallone-the-expendables-4-sequel-john-thompson-jason-statham-randy-couture-instagram-1202242598/|title=Sylvester Stallone Hints That 'The Expendables 4' Is Back on Track|last=Ramos|first=Dino-Ray|date=13 January 2018|website=Deadline Hollywood|access-date=20 April 2018}}&lt;/ref&gt;[[File:Cannes 2014 2.jpg|thumb|upright=1.3|Statham (centre) at the [[2014 Cannes Film Festival]]|left]]In 2013, Statham had a cameo appearance at the end of ''[[Fast &amp; Furious 6]]'' as [[Deckard Shaw|the brother]] of the film's antagonist Owen Shaw ([[Luke Evans]]).&lt;ref&gt;{{cite magazine|url=https://www.ew.com/article/2013/05/25/fast-furious-6-spoilers-post-credit-ending|title='Fast &amp; Furious 6' spoilers ending|first=Darren|last=Franich|date=25 May 2013|magazine=Entertainment Weekly|access-date=21 June 2015}}&lt;/ref&gt; He reprised the character, this time as the main antagonist, in ''[[Furious 7]]'', which was released in April 2015.&lt;ref&gt;{{cite web|url=https://screenrant.com/fast-furious-6-7-villain-post-credits-scene-spoilers/|title='Fast &amp; Furious 6' Post-Credits Scene &amp; 'Fast &amp; Furious 7' Villain Revealed|last=Keyes|first=Rob|website=Screenrant|date=14 February 2013|access-date=16 February 2013}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://screenrant.com/fast-furious-7-jason-statham/|title=Jason Statham Talks ''Fast &amp; Furious 7''|last=Frappier|first=Rob|date=June 2013|website=Screen Rant|access-date=26 August 2013}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.ew.com/article/2015/04/02/furious-7-statham-shaw-crank-3|title=Jason Statham talks 'Furious 7'...and 'Crank 3'|first=Darren|last=Franich|date=2 April 2015|work=Entertainment Weekly's EW.com|access-date=21 June 2015}}&lt;/ref&gt; He also starred opposite [[James Franco]] in the thriller ''[[Homefront (2013 film)|Homefront]]'', written by [[Sylvester Stallone]],&lt;ref&gt;{{cite web|url=https://collider.com/the-expendables-3-jason-statham-jackie-chan/|title=Jason Statham Hopes Jackie Chan Will Join Him in THE EXPENDABLES 3; Updates on HOMEFRONT Written by Sylvester Stallone|website=[[Collider (website)|Collider]]|access-date=18 November 2018|date=15 January 2013}}&lt;/ref&gt; and headlined the British thriller ''[[Hummingbird (film)|Hummingbird]]''. The latter film was praised by critics for pushing Statham's acting abilities to new heights. His &quot;attempt to develop his 'brand' by trying more adventurous parts&quot; noted by ''The Guardian''&lt;nowiki/&gt;'s [[Mark Kermode]], &quot;[broadened] his dramatic palette&quot;.&lt;ref&gt;{{Cite web|url=https://www.theguardian.com/film/2013/jun/30/hummingbird-jason-statham-review|title=Hummingbird – review|last=Kermode|first=Mark|date=29 June 2013|website=The Guardian|access-date=15 January 2018}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.theguardian.com/film/2013/jul/06/jason-statham-hummingbird-british-action|title=Jason Statham breaks into new acting territory with Hummingbird|last=Thorpe|first=Vanessa|date=6 July 2013|website=The Guardian|access-date=15 January 2018}}&lt;/ref&gt; Statham made a cameo in the 2014 music video ''[[Summer (Calvin Harris song)|Summer]]'' of [[Calvin Harris]] as one of the car racers. In 2014, he returned as Lee Christmas in ''[[The Expendables 3]].'' Although critically panned,&lt;ref&gt;{{Cite web|url=https://www.theguardian.com/film/2014/aug/17/the-expendables-3-review-macho-sparring-sylvester-stallone|title=The Expendables 3 review – Sly Stallone and co return for another macho bout of sparring|last=Barnes|first=Henry|date=16 August 2014|website=The Guardian|access-date=15 January 2018}}&lt;/ref&gt; the film would go on to gross $215 million against a $90 million budget.&lt;ref&gt;{{Cite web|url=https://www.boxofficemojo.com/movies/?id=expendables3.htm|title=The Expendables 3 (2014) – Box Office Mojo|website=boxofficemojo.com|access-date=15 January 2018}}&lt;/ref&gt;

589 Statham's impact on the action-thriller genre has been seen by Gabbatt as a replacement of the same undertaken by [[Arnold Schwarzenegger]], [[Sylvester Stallone]], and [[Jean-Claude Van Damme]] during their runs as headliners throughout the 1980s and 1990s.&lt;ref name=&quot;:02&quot;/&gt; Statham himself cites Stallone, [[Bruce Lee]], [[Paul Newman]], [[Steve McQueen]], and [[Clint Eastwood]] as his inspirations.&lt;ref name=&quot;:12&quot;&gt;{{Cite news|url=https://www.mensfitness.com/life/entertainment/jason-statham-in-his-own-words|title=Jason Statham in His Own Words|work=Men's Fitness|access-date=9 January 2018}}&lt;/ref&gt;

590 |&lt;ref&gt;{{cite web |last1=White |first1=James |title=Jason Statham Starring In David Ayer's Levon's Trade, Written By Sylvester Stallone |url=https://www.empireonline.com/movies/news/jason-statham-starring-in-david-ayers-levons-trade-written-by-sylvester-stallone/ |website=Empire |accessdate=13 May 2024 |date=29 October 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last1=Wiseman |first1=Andreas |title=Amazon MGM Lands Jason Statham Action-Thriller 'Levon's Trade' With David Ayer Directing Sylvester Stallone Script; Wide U.S. Theatrical Release Lined Up |url=https://deadline.com/2024/01/jason-statham-david-ayer-sylvester-stallone-amazon-levons-trade-movie-deal-1235805593/ |website=Deadline |accessdate=13 May 2024 |date=26 January 2024}}&lt;/ref&gt;

591 [[Sylvester Stallone]] was originally considered for the part of Foley.&lt;ref&gt;{{cite web |url=https://www.cinemablend.com/news/1544540/why-sylvester-stallones-beverly-hills-cop-movie-never-happened |title=Sylvester Stallone turns down Beverly Hills Cop Script according to book |last=O'Connell |first=Sean |date=August 11, 2016 |access-date=August 11, 2016 |archive-date=August 12, 2016 |archive-url=https://web.archive.org/web/20160812180902/http://www.cinemablend.com/news/1544540/why-sylvester-stallones-beverly-hills-cop-movie-never-happened |url-status=live }}&lt;/ref&gt; Stallone gave the script a dramatic rewrite, removing all the story's humor and turning the film back into a standard action movie.&lt;ref name=&quot;:0&quot;&gt;{{Cite book |last=Fleming |first=Charles |url=https://archive.org/details/highconceptdonsi0000flem |title=High Concept: Don Simpson and the Hollywood Culture of Excess |publisher=Bloomsbury Publishing |year=1998 |isbn=9780747536116 |location=London, England |pages=56 |language=English}}&lt;/ref&gt;&lt;ref name=&quot;CBR&quot;/&gt; In one of the previous drafts written for Stallone, the character of Billy Rosewood was called &quot;Siddons&quot; and was killed off half-way through the script during one of the action scenes.&lt;ref&gt;{{cite web | url= https://www.empireonline.com/features/recasting-blockbusters/ | title= Re-Cast: Five Blockbusters Completely Changed For Their Star | magazine= Empire Magazine | access-date= 2010-11-13 | archive-date= September 24, 2015 | archive-url= https://web.archive.org/web/20150924063839/http://www.empireonline.com/features/recasting-blockbusters/ | url-status= live }}&lt;/ref&gt; Stallone had renamed the lead character to Axel Cobretti, with the character of Michael Tandino being his brother and Jenny Summers playing his love interest.&lt;ref name=&quot;CBR&quot;/&gt;

592 Stallone has said that his script for ''Beverly Hills Cop'' would have &quot;looked like the opening scene from ''[[Saving Private Ryan]]'' on the beaches of Normandy. Believe it or not, the finale was me in a stolen [[Lamborghini]] playing chicken with an oncoming freight train being driven by the ultra-slimy bad guy.&quot;&lt;ref name=&quot;CBR&quot;/&gt; Producer Don Simpson let it be known they didn't want to move forward with Stallone's revisions; since Stallone wasn't willing to negotiate the rewrite, Simpson asked writer Charles &quot;Chip&quot; Proser if he could return the script to previous iteration, while leaving most of Stallone's character revisions intact. However Proser found the task (and turnaround time) preposterous.&lt;blockquote&gt;&lt;nowiki/&gt;&quot;''They offered me the rewrite when it was nothing more than Sylvester Stallone and an exotic gun-which was pretty ridiculous''&quot;, remembered screenwriter Chip Proser, who would later write [an uncredited rewrite of] Simpson and Bruckheimer's ''[[Top Gun]]''.&lt;ref name=&quot;:0&quot; /&gt;&lt;/blockquote&gt;According to co-producer Don Simpson, Stallone's new script spent too much time on the star '''soaping down his muscles''.'&lt;ref&gt;{{Cite book |last=Daly |first=Marsha |url=https://archive.org/details/sylvesterstallon0000daly |title=Sylvester Stallone |publisher=St. Martins Press |year=1986 |isbn=9780312903589 |edition=mass market |location=New York City, New York |pages=134 |language=English}}&lt;/ref&gt; Stallone ultimately dropped out two weeks before filming was to start, ostensibly to concentrate on his next picture, the 1984 film [[Rhinestone (film)|''Rhinestone'']].&lt;ref name=&quot;:0&quot; /&gt; Stallone later used the bulk of these ideas as the basis for the 1986 film ''[[Cobra (1986 film)|Cobra]]''.&lt;ref name=&quot;Fleming 1998 56-57&quot;&gt;{{Cite book |last=Fleming |first=Charles |url=https://archive.org/details/highconceptdonsi0000flem |title=High Concept: Don Simpson and the Hollywood culture of excess |publisher=Bloomsbury |year=1998 |isbn=9780385486941 |location=London, England |pages=56-57 |language=English}}&lt;/ref&gt;

593 Michael taunts the fans into chasing him, once they realize it is him. Michael steals a bicycle which then turns into a motorcycle once he hits the highway in order to flee, with the fans and the [[paparazzi]] (now also appearing in Claymation form) giving chase in various other, more cartoonish vehicles. During the chase, he morphs into other celebrities, namely [[Sylvester Stallone]], [[Tina Turner]] and [[Pee-wee Herman]], with the bike also briefly morphing (into a jackhammer, stop sign, waterski, and jet pack) in repeatedly unsuccessful attempts to throw the fans off.

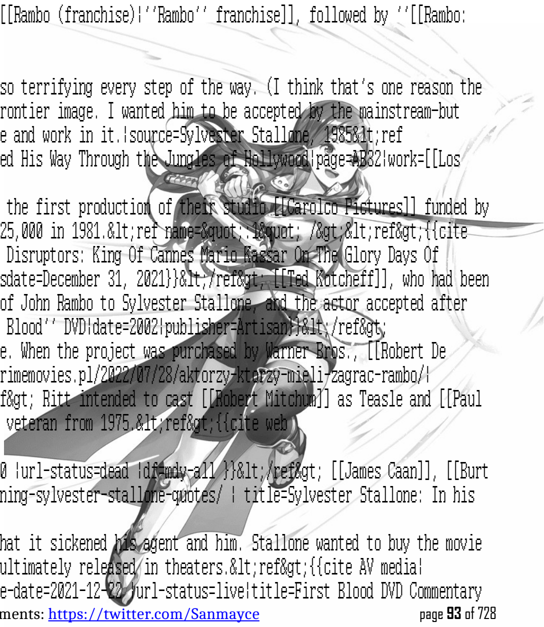
594 In 1991, Landis directed [[Sylvester Stallone]] in ''[[Oscar (1991 film)|Oscar]]'', based on a {{Ill|Claude Magnier|fr}} stage play. ''Oscar'' recreates a 1930s-era film, including the gestures along with bit acts and with some slapstick, as an homage to old Hollywood films.&lt;ref&gt;{{Citation|last=Landis|first=John|title=Oscar|date=April 26, 1991|url=https://www.imdb.com/title/tt0102603/?ref_=nm_flmg_dr_26|type=Comedy, Crime|others=Sylvester Stallone, Ornella Muti, Peter Riegert, Chazz Palminteri|publisher=Joseph S. Vecchio Entertainment, Silver Screen Partners IV, Touchstone Pictures|access-date=December 2, 2020}}&lt;/ref&gt; In 1992, Landis directed ''[[Innocent Blood (film)|Innocent Blood]]'', a [[horror film|horror]]-[[crime film]]. In 1994, Landis directed Eddie Murphy in ''[[Beverly Hills Cop III]]'', their third collaboration following ''[[Trading Places]]'' and ''[[Coming to America]]''. In 1996, he directed ''[[The Stupids (film)|The Stupids]]'' and then returned to Universal to direct ''[[Blues Brothers 2000]]'' in 1998 with [[John Goodman]] and, for the fifth time in a Landis film, [[Dan Aykroyd]], who also appeared in Landis' film ''[[Susan's Plan]]'', released that same year. None of the above six films scored well with critics or audiences.

595 De Carlo's later films included ''[[American Gothic (1988 film)|American Gothic]]'' (1988), for which she won the Best Actress Award from [[Fantafestival|International Science Fiction and Fantasy Film Show (Fantafestival)]]; ''[[Cellar Dweller]]'' (1988); and ''[[Mirror, Mirror (1990 film)|Mirror Mirror]]'' (1990). She had a supporting role as the title character's Aunt Rosa in the [[Sylvester Stallone]] comedy ''[[Oscar (1991 film)|Oscar]]'' (1991). Aunt Rosa is present when Oscar's father, played by [[Kirk Douglas]], extracts &quot;a deathbed promise&quot; from his son. Of her role, De Carlo said, &quot;Mine is a small part-but funny.&quot;&lt;ref name=&quot;In Search of ... Yvonne De Carlo&quot;&gt;{{cite news|last1=Broeske|first1=Pat H.|title=In Search of ... Yvonne De Carlo|url=https://www.latimes.com/archives/la-xpm-1991-01-13-ca-155-story.html|access-date=January 22, 2017|work=[[Los Angeles Times]]|date=January 13, 1991|archive-date=February 2, 2017|archive-url=https://web.archive.org/web/20170202021135/http://articles.latimes.com/1991-01-13/entertainment/ca-155_1_carlo-yvonne-tv|url-status=live}}&lt;/ref&gt;

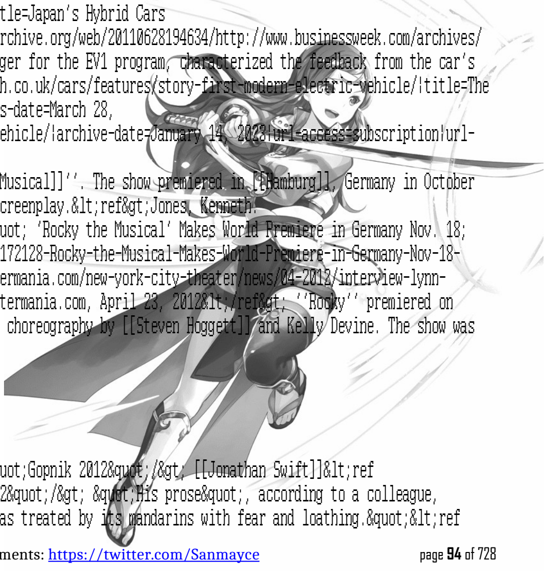
596 * [[Sylvester Stallone]] as [[Terminator (character)|the Terminator]] on a poster promoting ''Terminator 2: Judgment Day''. This references the [[Schwarzenegger-Stallone rivalry]].

597 Beatty had, for a couple of years, been going to [[OrlandoCon]], an annual comic book convention held in [[Orlando, Florida]]. Beatty and his long-time friend Craig Zablo (creator of Stallonezone, a [[Sylvester Stallone]] [[fansite]]), would go to OrlandoCon and make a weekend of it. On this particular occasion,{{when|date=September 2016}} Beatty took some art samples to show professional attendees, comics artists such as [[Pat Broderick]] and [[Bob McLeod (comics)|Bob McLeod]] (who were living on the west coast of Florida in Tampa).{{citation needed|date=September 2016}} This was also the con where Beatty met [[AC Comics]] publisher and artist [[Bill Black (comics)|Bill Black]]. Beatty showed samples to Black and was offered ink work on the spot; he would get to ink Black's pencils at a rate of $7.00 per page.{{citation needed|date=September 2016}}

```
# Miss Lidia's Makeover to the Stars &quot;Sylvester Stallone&quot;
* [[Sylvester Stallone]] - Star of the Year Award
* [[Sylvester Stallone]] - Star of Stars Award
*[[Sylvester Stallone]] - Career Achievement Award
Funk married his wife Vicky Ann on August 14, 1965. They had two daughters together. For many years, Terry and Vicky owned a ranch in [[Canyon, Texas]], which they later sold. Vicky died on March 29, 2019, as confirmed by [[Ted DiBiase]] on [[Twitter]].&lt;ref&gt;{{cite tweet|number=1112060527163908098|user=MDMTedDiBiase|title=My heart is broken and goes out to my good friend and mentor, Terry Funk. Terry's wife, Vickie, passed away yesterd…&lt;!-- full text of tweet (excluding links) added by TweetCiteBot. This may be better truncated. --&gt; |date=March 30, 2019}}&lt;/ref&gt; Funk was close friends with [[NFL]] player [[John Ayers]]. Funk was also close friends with [[Sylvester Stallone]]. On September 12, 2016, Funk underwent surgery for an inguinal hernia, and was supposed to be resting and in bed for a couple of weeks, but chose to attend Tommy Dreamer's House of Hardcore shows.&lt;ref&gt;{{cite web |url=http://www.f4wonline.com/other-wrestling/terry-funk-alive-and-well-texas-after-exhausting-weekend-221171|title=Terry Funk has tough week |last=Meltzer |first=Dave |date=September 19, 2016 |access-date=September 19, 2016}}&lt;/ref&gt;
The game was also inspired by the [[action film]] ''[[Rambo: First Blood Part II]]'' (1985), released as ''Rambo: Ikari no Dasshutsu'' (&quot;Rambo: The Furious Escape&quot;) in Japan. Obada had originally intended the game to be an official licensed adaptation of ''Rambo'', but SNK were initially unable to acquire the rights to the film. This resulted in the game's title being shortened to ''Ikari'', referencing part of the film's Japanese title. After the game made its North American debut at an arcade game expo, they got in touch with the film's star [[Sylvester Stallone]], who was friends with SNK's president, about acquiring the rights to the film, but it was too late by that point, as the game had already become popularly known by its Japanese title ''Ikari'' among arcade players in Japan and North America, leading to the game being officially released as ''Ikari Warriors'' in North America.&lt;ref name=&quot;Continue&quot;/&gt;
[[Sylvester Stallone]], whose role as [[John Rambo]] had an influence on the game, owned an ''Ikari Warriors'' arcade cabinet in the 1980s. The future [[mixed martial arts]] (MMA) champion [[Kazushi Sakuraba]] (later famous for defeating the [[Gracie family]] of [[Brazilian jiu-jitsu]] fighters) was also a fan of ''Ikari'' in his high school years.&lt;ref name=&quot;Continue&quot;/&gt;
After leaving ''NYPD Blue'', Brenneman made her way into film. In 1995, she appeared in ''[[Bye Bye Love (film)|Bye Bye Love]]'', ''[[Casper (film)|Casper]]'', and the critically acclaimed crime drama ''[[Heat (1995 film)|Heat]]''.&lt;ref name=yahoo/&gt; In 1996, Brenneman had the female lead role opposite [[Sylvester Stallone]] in disaster thriller ''[[Daylight (1996 film)|Daylight]]'', and also starred in another thriller, ''[[Fear (1996 film)|Fear]]''. In 1997, she played the leading role in the independent drama film ''[[Nevada (1997 film)|Nevada]]''.&lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/v154851 |title=Nevada (1997) - Trailers, Reviews, Synopsis, Showtimes and Cast |publisher=AllMovie |access-date=2013-12-30}}&lt;/ref&gt; The next year she starred in ''[[Your Friends &amp; Neighbors]]'' directed by [[Neil LaBute]].&lt;ref&gt;{{Citation|title=Your Friends &amp; Neighbors|url=https://www.rottentomatoes.com/m/your_friends_and_neighbors/|work=Rotten Tomatoes|access-date=June 5, 2017}}&lt;/ref&gt; In 1998-1999 season, she returned to television with a recurring role on the NBC comedy, ''[[Frasier]]'', as Faye Moskowitz.&lt;ref name=amy/&gt;
Overseas, the film initially received a mixed reaction in contemporary English-language reviews. [[Kim Newman]] wrote, in the ''[[Monthly Film Bulletin]]'', that ''Police Story'' starts well with its car chase described as an &quot;astonishing set-piece&quot;, but that &quot;once the mix of realistic settings and fantasy action seems to have been established, the film falls back on Chan's clowning and turns into a slapstick comedy heavily dependent on cake-in-the-face jokes&quot;. The review concluded that the film &quot;still lacks much of the aesthetic appeal and occasional comic grace of the more traditional period kung fu films such as ''[[Project A (film)|Project A]]'', ''[[Drunken Master]]'', ''[[Snake in the Eagle's Shadow]]'' and ''[[Dragon Lord]]''&quot;.&lt;ref name=&quot;mfb-review&quot;&gt;{{cite journal|journal=[[Monthly Film Bulletin]]|title=Jingcha Gushi/Police Story|last=Newman|first=Kim|author-link=Kim Newman|volume=54|issue=637|pages=50-51|date=1987|publisher=[[British Film Institute]]|issn=0027-0407}}&lt;/ref&gt; [[Vincent Canby]] wrote in ''[[The New York Times]]'' that at a screening of the film at the [[New York Film Festival]] Chan was promoted as a hybrid of [[Buster Keaton]] and [[Clint Eastwood]], with Canby noting that Chan was &quot;more like a scaled-down, oriental [[Sylvester Stallone]], with energy and a willingness to smile fondly at himself&quot;. Canby also noted the excessive pie-in-the-face gags, and that Chan &quot;participates in several elaborately staged gun fights and car chases&quot; which were &quot;mildly amusing&quot; but not as amusing as the dubbed dialogue. Canby concluded that the film &quot;is of principal interest as a souvenir of another culture&quot;.&lt;ref name=&quot;nyt&quot;&gt;{{cite web|title=Film Festival; 'Jackie Chan's Police Story'|website=[[The New York Times]]|date=26 September 1987|url=https://www.nytimes.com/1987/09/26/movies/film-festival-jackie-chan-s-police-story.html|access-date=25 August 2015|first=Vincent|last=Canby|author-link=Vincent Canby|archive-date=11 October 2015|archive-url=https://web.archive.org/web/20151011184203/http://www.nytimes.com/1987/09/26/movies/film-festival-jackie-chan-s-police-story.html|url-status=live}}&lt;/ref&gt;
* The scene where Chan stops a bus inspired a similar scene in the [[Sylvester Stallone]] and [[Kurt Russell]] film ''[[Tango &amp; Cash]]''.&lt;ref name=&quot;Donovan&quot;/&gt;
He appeared with [[Goldie Hawn]] in ''[[Protocol (film)|Protocol]]'' (1984). In 1988 he played the father in ''[[Some Girls (film)|Some Girls]],'' with [[Jennifer Connelly]] and [[Patrick Dempsey]]. In 1993, he performed in the movie ''[[Demolition Man (film)|Demolition Man]]'' with [[Sylvester Stallone]].
She also co-starred with [[Ed O'Neill]] in the [[John Hughes (filmmaker)|John Hughes]]-written comedy ''[[Dutch (1991 film)|Dutch]]'' (1991), and starred in ''[[Stop! Or My Mom Will Shoot]]'' (1992) as the police detective/love interest of [[Sylvester Stallone]]'s character. In 1995, she was nominated for an Academy Award for her 1994 live-action short ''On Hope,'' starring [[Annette O'Toole]]; the film was Williams's directorial debut. In 1997, she played a domineering lesbian in the independent comedy ''[[Little City]]'' with [[Jon Bon Jovi]], and an hysterical publishing editor in ''[[Just Write]]'' with [[Jeremy Piven]]. In 2005, she appeared in the [[Drew Barrymore]]-[[Jimmy Fallon]] baseball comedy ''[[Fever Pitch]]''.
* [[Sage Stallone]] (1976-2012), actor and son of Sylvester Stallone&lt;ref&gt;{{cite news|title=Sage Stallone, Sylvester Stallone's Son, Found Dead in Hollywood Hills Apartment: Report|first=Simone|last=Wilson|date=July 13, 2012|journal=LAWeekly|url=http://blogs.laweekly.com/informer/2012/07/sage_stallone_sylvester_stallones_son_found_dead_hollywood_hills_apartment.php|access-date=April 30, 2013|archive-url=https://web.archive.org/web/20130919071943/http://blogs.laweekly.com/informer/2012/07/sage_stallone_sylvester_stallones_son_found_dead_hollywood_hills_apartment.php|archive-date=September 19, 2013|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite news|journal=New York Post|title=Sylvester Stallone's son found dead; cause of death believed to be pill overdose|first1=Emily|last1=Smith|first2=Ian|last2=Mohr|date=July 14, 2012|url=http://www.nypost.com/p/news/national/sylvester_stallone_son_found_dead_cVP2gGx7BBzgeNax2Dm3JL}}&lt;/ref&gt;
Chuck Wepner was a heavyweight boxer in the 1960s-70s whose notable loss to [[Muhammad Ali]] in 1975 became the inspiration for [[Sylvester Stallone|Sylvester Stallone's]] Oscar winning movie [[Rocky]]. Stallone named the main character for his 1989 film [[Lock Up (1989 film)|&quot;Lock Up&quot;]] after Wepner. Stallone met with Wepner during filming at the prison, when he was an inmate.&lt;ref&gt;{{Cite web |last=Christopher |first=Michael |title=30 Years Ago: 'Lock Up' Takes Stallone's 'Rocky' Formula to Jail |url=https://ultimateclassicrock.com/stallone-lock-up/ |access-date=2022-07-04 |website=Ultimate Classic Rock |language=en}}&lt;/ref&gt; Wepner was also the inspiration for the major motion pictures, [[Chuck (film)|''Chuck (2016)'']] and ''[[The Brawler|The Brawler (2019)]]''.
* [[Lock Up (1989 film)|''Lock Up'' (1989)]] - Starring [[Sylvester Stallone]] and [[Donald Sutherland]]
Among the final revisions of the script was a change in location. The original script set the action in San Francisco, but the revision moved the setting to Los Angeles.&lt;ref name=&quot;Talbot&quot;&gt;Talbot (2006), p. 31-58&lt;/ref&gt; Winner said the film was &quot;the same, but different&quot;, from the original. &quot;That's what sequels are - ''[[Rocky II]]'', ''[[Rocky III]]'' - you don't see [[Sylvester Stallone]] move to the [[Congo Basin|Congo]] and become a nurse. Here the look of LA is what's different. Besides - rape doesn't date!&quot;&lt;ref name=&quot;death&quot;&gt;{{cite news|title=THE REINCARNATION OF A 'DEATH WISH'|author=Trombetta, Jim|work=Los Angeles Times|date=13 July 1981|page=g1}}&lt;/ref&gt;
Kotcheff directed the Canadian film ''[[Split Image (film)|Split Image]]'' (1982), then had his biggest success to date with the [[Sylvester Stallone]] movie ''[[First Blood]]'' (1982), the first in the [[Rambo (franchise)|Rambo series]]. He worked on another Vietnam-themed action movie ''[[Uncommon Valor]]'' (1983), then returned to Canada to make ''[[Joshua Then and Now (film)|Joshua Then and Now]]'' (1985), from the novel by [[Mordecai Richler]].
* [[Sylvester Stallone]]
* [[Sylvester Stallone]]
'''''First Blood''''' is a 1982 American [[action film]] directed by [[Ted Kotcheff]] and co-written by and starring [[Sylvester Stallone]] as [[Vietnam War]] veteran [[John Rambo]]. It co-stars [[Richard Crenna]] as Rambo's mentor Sam Trautman and [[Brian Dennehy]] as Sheriff Will Teasle. It is the first installment in the [[Rambo (franchise)|''Rambo'' franchise]], followed by ''[[Rambo: First Blood Part II]]''.
* [[Sylvester Stallone]] as [[John Rambo|John J. Rambo]]
{{Quote box |align=right |quoted=true |salign=center |width=20em |The original Rambo was so bloodthirsty... the story was so hard, so terrifying every step of the way. (I think that's one reason the book took so long to get done). What I did with Rambo was try to keep one foot in the Establishment and one foot in the outlaw or frontier image. I wanted him to be accepted by the mainstream-but also be a criminal. So he has some strong patriotic views-and he loves the system. He just doesn't like a lot of the people who live and work in it.|source=Sylvester Stallone, 1985&lt;ref name=&quot;:0&quot;&gt;{{Cite news|last=Broeske|first=Pat H.|date=1985-11-25|title=The Curious Evolution of John Rambo: How He Hacked His Way Through the Jungles of Hollywood|page=AB32|work=[[Los Angeles Times]]|publication-place=Los Angeles}}&lt;/ref&gt;}}
After [[Mario Kassar]] and [[Andrew G. Vajna]] of Anabasis Investments read the book, they got interested in doing an adaptation as the first production of their studio [[Carolco Pictures]] funded by &quot;in-house sources&quot;. They purchased the film rights from Warner Bros. for $375,000 and Sackheim and Kozoll's script for $125,000 in 1981.&lt;ref name=&quot;:1&quot; /&gt;&lt;ref&gt;{{cite web|url=https://deadline.com/2016/05/carolco-pictures-mario-kassar-cannes-interview-foxtrot-six-audition-1201752739/|title=Deadline Disruptors: King Of Cannes Mario Kassar On The Glory Days Of Carolco, Why Buying Arnie A Plane Made Sense &amp; Talking Vaginas|first=Ali|last=Jaafar|publisher=Deadline|date=May 12, 2016|accessdate=December 31, 2021}}&lt;/ref&gt; [[Ted Kotcheff]], who had been involved in the project in 1976, returned after Kassar and Vajna offered to finance one of his projects. Kotcheff offered the role of John Rambo to Sylvester Stallone, and the actor accepted after reading the script through in a weekend.&lt;ref name=&quot;drawing&quot;&gt;{{cite video|title=Drawing First Blood|location=''First Blood'' DVD|date=2002|publisher=Artisan}}&lt;/ref&gt;
Brooks originally wanted to cast [[Bette Davis]] as a psychiatrist and either [[Burt Lancaster]] or [[Lee Marvin]] as Sheriff Teasle. When the project was purchased by Warner Bros., [[Robert De Niro]] and [[Clint Eastwood]] were each considered for the role of Rambo.&lt;ref name=&quot;prime&quot;&gt;{{cite web|url=https://primemovies.pl/2022/07/28/aktorzy-ktorzy-mieli-zagrac-rambo/|title=Ten actors that almost portrayed John Rambo on screen|work=Prime Movies|date=July 28, 2022|access-date=July 29, 2022}}&lt;/ref&gt; Ritt intended to cast [[Robert Mitchum]] as Teasle and [[Paul Newman]] as Rambo. Pollack considered [[Steve McQueen]] but then rejected him because they considered him too old to play a Vietnam veteran from 1975.&lt;ref&gt;{{cite web |url=http://www.yuddy.com/celebrity/steve-mcqueen/bio |title=Steve Mcqueen Bio |publisher=Yuddy.com |access-date=July 18, 2010 |archive-url=https://web.archive.org/web/20100718224247/http://www.yuddy.com/celebrity/steve-mcqueen/bio |archive-date=July 18, 2010 |url-status=dead |df=mdy-all }}&lt;/ref&gt; [[James Caan]], [[Burt Reynolds]] and [[Robert Redford]] were also considered.&lt;ref&gt;{{cite web | url=https://www.empireonline.com/movies/features/evening-sylvester-stallone-quotes/ | title=Sylvester Stallone: In his own words | date=January 14, 2014 }}&lt;/ref&gt;
The first rough cut of the film was between three and three-and-a-half hours long. According to Sylvester Stallone, it was so bad that it sickened his agent and him. Stallone wanted to buy the movie and destroy it thinking that it was a career killer. After heavy re-editing, the film was cut down to 93 minutes; this version was ultimately released in theaters.&lt;ref&gt;{{cite AV media|url=https://www.youtube.com/watch?v=tGYN0DBRVjI |archive-url=https://ghostarchive.org/varchive/youtube/20211222/tGYN0DBRVjI |archive-date=2021-12-22 |url-status=live|title=First Blood DVD Commentary
```



```
    by Sylvester Stallone|date=July 19, 2014|work=YouTube}}{{cbignore}}&lt;/ref&gt; The ending used in the finished film was shot in March 1982, after the original one was deemed unsatisfactory.&lt;ref&gt;{{cite news|work=The Ottawa Citizen|location=Ottawa, Ontario, Canada|title=News|date=March 12, 1982}}&lt;/ref&gt;
623 On [[review aggregator]] [[Rotten Tomatoes]], the film has an 86% approval rating based on 49 reviews, with an average rating of 7.20/10. The site's critics consensus reads, &quot;Much darker and more sensitive than the sequels it spawned, ''First Blood'' is a thrilling survival adventure that takes full advantage of Sylvester Stallone's acting skills.&quot;&lt;ref&gt;{{cite web | url=https://www.rottentomatoes.com/m/first_blood |title=First Blood (1982) |website=[[Rotten Tomatoes]] |publisher=[[Fandango Media|Fandango]] |access-date=July 17, 2022 |archive-date=April 30, 2019 |archive-url=https://web.archive.org/web/20190430045234/https://www.rottentomatoes.com/m/first_blood |url-status=live }}&lt;/ref&gt; On [[Metacritic]], the film has a weighted average score of 61 out of 100 based on 15 critics, indicating &quot;generally favorable reviews&quot;.&lt;ref name=&quot;First Blood 1982: Reviews&quot;&gt;{{cite web |url=https://www.metacritic.com/movie/first-blood | title=First Blood Reviews |website=[[Metacritic]] |publisher=[[CBS Interactive]] |access-date=September 24, 2019 |archive-date=November 12, 2020 |archive-url=https://web.archive.org/web/20201112035558/https://www.metacritic.com/movie/first-blood |url-status=live }}&lt;/ref&gt;
624 On August 14, 2020, a cedar wood statue of Rambo was unveiled in [[Hope, British Columbia|Hope]], [[Canada]], 38 years after the film's release. Mayor Peter Robb, Canadian Member of Parliament Mark Strahl, and the statue's sculptor, Ryan Villers, attended the ceremony.&lt;ref&gt;{{cite web | url=https://bc.ctvnews.ca/sylvester-stallone-very-proud-of-new-rambo-carving-in-b-c-town-1.5069785 | title=Sylvester Stallone 'very proud' of new Rambo carving in B.C. town | first=Andrew | last=Weichel | date=August 18, 2020 | access-date=January 30, 2021 | archive-date=December 14, 2020 | archive-url=https://web.archive.org/web/20201214010240/https://bc.ctvnews.ca/sylvester-stallone-very-proud-of-new-rambo-carving-in-b-c-town-1.5069785 | url-status=live }}&lt;/ref&gt;
625 {{Sylvester Stallone}}
626 [[Category:Films with screenplays by Sylvester Stallone]]
627 Gray began her career in the 1960s in television commercials. In the 1970s, she appeared in numerous TV series before landing the role of Sue Ellen Ewing in 1978. After leaving ''[[Dallas (1978 TV series)|Dallas]]'' in 1989, she appeared opposite [[Sylvester Stallone]] in the 1991 film ''[[Oscar (1991 film)|Oscar]]''. From 1994 to 1995, she played a leading role in the [[Fox Broadcasting Company|Fox]] drama series ''[[Models Inc.]]'', and also starred in TV movies, including ''[[Moment of Truth: Why My Daughter?]]'' (1993) and ''[[Accidental Meeting (1994 film)|Accidental Meeting]]'' (1994). She went on to reprise the role of Sue Ellen in ''[[Dallas: J.R. Returns]]'' (1996), ''[[Dallas: War of the Ewings]]'' (1998), and in the [[TNT (U.S. TV network)|TNT]] series ''[[Dallas (2012 TV series)|Dallas]]'' (2012-2014), which continued the original series.
628 After ''Dallas'', Gray starred in the 1991 comedy film, ''[[Oscar (1991 film)|Oscar]]'' with [[Sylvester Stallone]] and then made guest appearances in British drama ''[[Lovejoy]]'' (starring her co-star and love interest from her final season on ''Dallas'', [[Ian McShane]]). She also starred in several made-for-TV movies, including 1991's ''The Entertainers'' (with [[Bob Newhart]]), ''[[Bonanza: The Return]]'' (1993), ''[[Moment of Truth: Why My Daughter]]'' (1993), and ''Accidental Meeting'' (1994).
629 The same year, Wagner released her first and only promotional single mononymously as Jasmin, the Christmas song &quot;[[Santa Claus Is Comin' to Town]]&quot;, an [[Eddie Cantor]] cover, which was only available exclusively at [[Tchibo]] stores.&lt;ref&gt;{{Cite web|url=https://www.laut.de/News/Jasmin-Wagner-Tchibo-Kaffee-mit-Bluemchen-Aroma-19-11-2001-822|title=Jasmin Wagner: Tchibo Kaffee mit Blümchen-Aroma - laut.de - News}}&lt;/ref&gt; Wagner has also ventured into acting making cameo appearances in several movies, most notably as a race car fan in the 2001 film ''[[Driven (2001 film)|Driven]]'', starring [[Sylvester Stallone]].&lt;ref&gt;{{cite web|url=https://www.imdb.com/title/tt0132245 |title=Driven (2001) - IMDb|website=[[IMDb]] }}&lt;/ref&gt;
630 Hedlund portrayed controversial U.S. government official [[Harry J. Anslinger]] in the [[biographical film]] ''[[The United States vs. Billie Holiday]]'' in 2021, directed by [[Lee Daniels]]; Anslinger was actually in his mid 50s to 60s during the setting of the film.&lt;ref name=&quot;2paragraphs.com&quot;&gt;[https://2paragraphs.com/2020/10/emma-robertss-boyfriend-plays-crazy-racist-in-lee-daniels-billie-holiday-biopic/ «Garrett Hedlund plays Harry J. Anslinger in Lee Daniels Billie Holiday Biopic»] - October 19, 2020.&lt;/ref&gt;&lt;ref&gt;[https://glittermagrocks.com/connect/2021/02/16/interview-garrett-hedlund-discusses-his-role-as-harry-anslinger-in-hulus-original-film-the-united-states-vs-billie-holiday-directed-by-lee-daniels/ «Garrett Hedlund Discusses His Role as Harry Anslinger in Hulu's Original Film, 'The United States vs. Billie Holiday'»] - February 16, 2021.&lt;/ref&gt; In September 2021, he stars in the first [[Stephen King]] [[podcast]] titled ''[[Strawberry Spring]]''.&lt;ref&gt;{{cite magazine|magazine=Rolling Stone|title=Stephen King Gets into the Podcast Game With a Series Based on 'Night Shift'|access-date=July 21, 2021|author=Ehrlich, Brenna|url=https://www.rollingstone.com/culture/culture-news/strawberry-spring-stephen-king-podcast-1199678/|date=July 20, 2021}}&lt;/ref&gt;&lt;ref name=&quot;Ehrlich, Brenna&quot;&gt;{{cite magazine|magazine=Rolling Stone|title=Garrett Hedlund Talks Starring in Stephen King's 'Strawberry Spring' Podcast|access-date=August 25, 2021|author=Ehrlich, Brenna|url=https://www.rollingstone.com/culture/culture-news/garrett-hedlund-stephen-king-strawberry-spring-trailer-1216260/|date=August 25, 2021}}&lt;/ref&gt; Hedlund's first single not associated with an acting role, &quot;The Road&quot;, was digitally released on January 21, 2022.&lt;ref name=&quot;people.com&quot;&gt;[https://people.com/country/garrett-hedlund-joins-instagram-and-releases-new-single-following-split-from-emma-roberts/ «Garrett Hedlund Joins Instagram and Releases First Single 'The Road'»] - January 22, 2022.&lt;/ref&gt; That same year, he co-starred with [[Sylvester Stallone]] in the [[Paramount+]] series ''[[Tulsa King]]'', playing bartender and ex-bull rider Mitch Keller.&lt;ref&gt;[https://www.nerdsandbeyond.com/2022/11/17/six-garrett-hedlund-films-series-you-can-stream-right-now/ «Six Garrett Hedlund Films &amp; Series You Can Stream Right Now»] - November 17, 2022.&lt;/ref&gt;
631 | &lt;ref&gt;{{cite web |last1=Otterson |first1=Joe |title=Sylvester Stallone Paramount+ Series ''Tulsa King'' Casts Garrett Hedlund (EXCLUSIVE) |url=https://variety.com/2022/tv/news/sylvester-stallone-paramount-plus-tulsa-king-cast-garrett-hedlund-1235264069/ |website=[[Variety (magazine)|Variety]] |access-date=11 May 2022 |date=May 11, 2022}}&lt;/ref&gt;
632 | &lt;ref&gt;[https://www.thelist.com/1288361/sylvester-stallone-family-discuss-the-family-stallone-living-the-spotlight/ «Sylvester Stallone And His Family On The Family Stallone And Living In The Spotlight - Exclusive Interview»] - May 17, 2023.&lt;/ref&gt;
633 | ''[[Cliffhanger (film)|Cliffhanger]]'' || Gabe Walker || [[Sylvester Stallone]] || 1997 NTV edition || &lt;ref&gt;{{cite web |url= https://www.tv-asahi.co.jp/nichiyou_2012/bk/data/01648.html|title= クリフハンガー|work= TV Asahi|accessdate= February 20, 2021|archiveurl= https://web.archive.org/web/20160915040449/https://www.tv-asahi.co.jp/nichiyou_2012/bk/data/01648.html|archivedate= September 15, 2016}}&lt;/ref&gt;
634 | ''[[D-Tox]]'' || Jake Malloy || Sylvester Stallone || || &lt;ref&gt;{{cite web |url= https://www.star-ch.jp/channel/detail.php?movie_id=15173|title= D-TOX [吹] |access-date= March 13, 2019| work= Star Channel}}&lt;/ref&gt;
635 Among his acting roles are bank robber Reed Youngblood in ''[[Dillinger (1973 film)|Dillinger]]'' (1973); a shouting police captain in ''[[48 Hrs.]]'' (1982), a role he later parodied in ''[[Last Action Hero]]'' (1993) and ''[[Loaded Weapon 1]]'' (1993); the history teacher, Mr. Teasdale, in ''[[Red Dawn]]'' (1984); [[James Bond]]'s friend Sharkey in ''[[Licence to Kill]]'' (1989); and former professional [[boxing]] champion Harry Noble in ''[[Batteries Not Included]]'' (1987). He also appeared opposite [[Sylvester Stallone]] in 4 films: ''[[F.I.S.T. (film)|F.I.S.T.]]'' (1978), ''[[Paradise Alley]]'' (1978), ''[[Rocky II]]'' (1979), and ''[[Lock Up (1989 film)|Lock Up]]'' (1989), as well as a protective trucker in ''[[The Wizard (1989 film)|The Wizard]]'' (1989).
636 Similar to the PrEView program, lessees were pre-screened by GM, with only residents of Southern California and Arizona initially eligible for participation.&lt;ref&gt;{{cite news|url=https://www.nytimes.com/1997/05/07/business/gm-is-trying-to-make-a-go-of-its-electric-car.html|title=G.M. Is Trying to Make a Go Of Its Electric Car|newspaper=[[The New York Times]]|date=May 7, 1997|access-date=March 28, 2024|first=Andrea|last=Adelson|archive-url=https://web.archive.org/web/20230120210921/https://www.nytimes.com/1997/05/07/business/gm-is-trying-to-make-a-go-of-its-electric-car.html|archive-date=January 20, 2023|url-status=live|url-access=subscription}}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=https://www.latimes.com/archives/la-xpm-1996-11-26-fi-3165-story.html| title=GM's EV1 Gears Up Amid Charge of the Ad Brigade|date=November 26, 1996|first=Donald W.|last=Nauss|newspaper=[[Los Angeles Times]]|access-date=March 28, 2024|archive-url=https://web.archive.org/web/20221205151056/https://www.latimes.com/archives/la-xpm-1996-11-26-fi-3165-story.html|archive-date=December 5, 2022|url-access=subscription|url-status=live}}&lt;/ref&gt; Leasing rates for the EV1 ranged from $399 to $549 a month.&lt;ref&gt;{{Cite magazine |last=Naughton|first=Keith|date=December 15, 1997 |title=Detroit: It Isn't Easy Going Green |url=http://www.businessweek.com/archives/1997/b3557013.arc.htm#B3557019 |url-status=dead |magazine=[[Business Week]] |archive-url=https://web.archive.org/web/20110628194634/http://www.businessweek.com/archives/1997/b3557013.arc.htm#B3557019 |archive-date=June 28, 2011 |access-date=March 28, 2024}}&lt;/ref&gt; The car's debut was marked by a significant media event, featuring an [[United States dollar|US$]]8 million promotional campaign incorporating prime-time TV commercials, billboards, a dedicated website, and an appearance at the premiere of the [[Sylvester Stallone]] film [[Daylight (1996 film)|Daylight]]. Among the initial lessees were notable figures such as celebrities, executives, and politicians. At the release event, 40 EV1 leases were signed, with GM anticipating leasing 100 cars by year's end. Deliveries began on December 5, 1996.&lt;ref name=&quot;LAtimes&quot;/&gt; In the first year on the market, GM leased just 288 cars.&lt;ref&gt;{{Cite magazine |last=Thornton|first=Emily |date=December 15, 1997 |title=Japan's Hybrid Cars |url=http://www.businessweek.com/archives/1997/b3557013.arc.htm |url-status=dead |magazine=Business Week |archive-url=https://web.archive.org/web/20110628194634/http://www.businessweek.com/archives/1997/b3557013.arc.htm |archive-date=June 28, 2011 |access-date=March 28, 2024}}&lt;/ref&gt; But in 1999 Ken Stewart, the brand manager for the EV1 program, characterized the feedback from the car's drivers as &quot;wonderfully-maniacal loyalty&quot;.&lt;ref name=&quot;TheTelegraphII&quot;&gt;{{cite news|url=https://www.telegraph.co.uk/cars/features/story-first-modern-electric-vehicle/|title=The story of the first 'modern' electric vehicle|first=Andrew|last=English|newspaper=[[The Daily Telegraph]]|date=January 1, 2022|access-date=March 28, 2024|archive-url=https://web.archive.org/web/20230114155523/https://www.telegraph.co.uk/cars/features/story-first-modern-electric-vehicle/|archive-date=January 14, 2023|url-access=subscription|url-status=live}}&lt;/ref&gt;{{sfn|Fletcher|2011|p=82}}
637 After writing three shows for Lincoln Center Theatre, Flaherty and Ahrens next returned to Broadway with the musical ''[[Rocky the Musical]]''. The show premiered in [[Hamburg]], Germany in October 2012. The musical has a book by [[Thomas Meehan (writer)|Thomas Meehan]] and [[Sylvester Stallone]], based on Stallone's original screenplay.&lt;ref&gt;Jones, Kenneth. [http://www.playbill.com/news/article/172128-Rocky-the-Musical-Makes-World-Premiere-in-Germany-Nov-18-American-Drew-Sarich-Stars &quot; 'Rocky the Musical' Makes World Premiere in Germany Nov. 18; American Drew Sarich Stars&quot;] {{webarchive|url=https://web.archive.org/web/20121119102059/http://www.playbill.com/news/article/172128-Rocky-the-Musical-Makes-World-Premiere-in-Germany-Nov-18-American-Drew-Sarich-Stars|date=2012-11-19}} playbill.com, November 18, 2012&lt;/ref&gt;&lt;ref&gt;Orlando, Nick. [http://www.theatermania.com/new-york-city-theater/news/04-2012/interview-lynn-ahrens-and-stephen-flaherty-continu_55047.html &quot;INTERVIEW: Lynn Ahrens and Stephen Flaherty Continue to Journey On&quot;] theatermania.com, April 23, 2012&lt;/ref&gt; ''Rocky'' premiered on Broadway at the [[Winter Garden Theatre]], officially opening on March 13, 2014. The musical was directed by [[Alex Timbers]], with choreography by [[Steven Hoggett]] and Kelly Devine. The show was nominated for 4 Tony Awards and 7 Drama Desk Awards including Outstanding Musical.
638     <redirect title="Sylvester Stallone" />
639       <text bytes="32" sha1="p7gz58bc0f4qqxa7url52a8fje2gfcq" xml:space="preserve">#REDIRECT [[Sylvester Stallone]]</text>
640     <redirect title="Sylvester Stallone" />
641       <text bytes="32" sha1="p7gz58bc0f4qqxa7url52a8fje2gfcq" xml:space="preserve">#REDIRECT [[Sylvester Stallone]]</text>
642 Hughes's critical prose, vivid in both praise and indignation, has been compared to that of [[George Bernard Shaw]],&lt;ref name=&quot;Gopnik 2012&quot;/&gt; [[Jonathan Swift]]&lt;ref name=&quot;Lacayo 2012&quot;/&gt; and [[William Shakespeare]].&lt;ref name=&quot;McNay 2012&quot;/&gt;&lt;ref name=&quot;Lacayo 2012&quot;/&gt; &quot;His prose&quot;, according to a colleague, &quot;was lithe, muscular and fast as a bunch of fives. He was incapable of writing the jargon of the art world, and consequently was treated by its mandarins with fear and loathing.&quot;&lt;ref
```



name=&quot;McNay 2012&quot;/&gt; In different moods he could write that &quot;[[Julian Schnabel|Schnabel's]] work is to painting what [[Sylvester Stallone|Stallone's]] is to acting: a lurching display of oily pectorals,&quot;&lt;ref name=&quot;Lacayo 2012&quot;/&gt; as well as conclude that [[Antoine Watteau]] &quot;was a connoisseur of the unplucked string, the immobility before the dance, the moment that falls between departure and nostalgia.&quot;&lt;ref name=&quot;McNay 2012&quot;/&gt;

643 His most popular [[Cinema of the United States|Hollywood]] releases are ''[[Maria's Lovers]]'' (1984), ''[[Runaway Train (film)|Runaway Train]]'' (1985), based on a script by Japanese director [[Akira Kurosawa]] (who had written it as an adaptation of Dostoevsky's &quot;[[The House of the Dead (novel)|House of the Dead]]&quot;&lt;ref&gt;{{Cite magazine |date=2020-12-21 |title=&quot;Dear Comrades!&quot; Is Andrei Konchalovsky's Masterpiece |url=https://www.newyorker.com/magazine/2021/01/04/dear-comrades-is-andrei-konchalovskys-masterpiece |access-date=2023-03-06 |magazine=The New Yorker |language=en-US}}&lt;/ref&gt;), and ''[[Tango &amp; Cash]]'' (1989), starring [[Sylvester Stallone]] and [[Kurt Russell]]. In 1985, [[Ned Tanen]] wanted to hire Konchalovsky to direct ''[[The Godfather Part III]]'', but the other executives at [[Paramount Pictures]] concluded the film could not be made without [[Francis Ford Coppola]] and [[Mario Puzo]]'s involvement.&lt;ref&gt;{{Cite web |title=The Godfather Part III |url=https://catalog.afi.com/Film/57313-8-MILLIONWAYSTODIE?sid=1f372f0c-5c41-4bb7-aceb-1f511ca1337c&amp;sr=15.708094&amp;cp=1&amp;pos=0 |access-date=2023-04-08 | website=AFI Catalog}}&lt;/ref&gt; In the 1990s, Konchalovsky returned to Russia, although he occasionally produced historical films for U.S. television, such as his adaption of ''[[The Odyssey (1997 miniseries)|The Odyssey]]'' (1997) and the award-winning [[remake]], ''[[The Lion in Winter (2003 film)|The Lion in Winter]]'' (2003). In a 2023 [[SAG-AFTRA Foundation]] interview, [[Emily Blunt]] stated that she auditioned for a role in the latter film, and said of Konchalovsky that &quot;he was vile... he was really horrible,&quot; and that he was &quot;really cruel during the audition and loved taking me down a peg or two.&quot; Blunt claimed that he promoted a &quot;very misogynistic sort of vibe, and I was a shell of my former self by the time I came out.&quot;&lt;ref&gt;{{Citation |title=Emily Blunt Career Retrospective {{!}} SAG-AFTRA Foundation Conversations |url=https://www.youtube.com/watch?v=ldTqJ2Sxf8w |access-date=2024-01-15 |language=en}}&lt;/ref&gt;

644 The film is based on a first draft by [[Zak Penn]], which allegedly impressed [[Steven Spielberg]] so much in its depiction of serial killers' elongated middle fingers that he went home and checked his children's hands.&lt;ref&gt;{{Cite web|url=http://www.aintitcool.com/node/12641|title=MORIARTY Captures the Shooting Draft of SUSPECT ZERO and Puts It Out of Its Misery!!}}&lt;/ref&gt; After it was sold to [[Universal Studios]] for $750,000, [[Cruise/Wagner Productions]] became attached to the film, with [[Tom Cruise]] and [[Paula Wagner]] producing. However the script was put onto the back burner after a deal to make the movie in 1997 with [[Sylvester Stallone]] fell through.&lt;ref&gt;{{Cite web|url=https://www.weeklystandard.com/jonathan-v-last/anatomy-of-a-murder|archive-url=https://web.archive.org/web/20181205172036/https://www.weeklystandard.com/jonathan-v-last/anatomy-of-a-murder|url-status=dead|archive-date=December 5, 2018|title=Anatomy of a Murder|date=10 September 2004}}&lt;/ref&gt; Cruise had intended to star in the film initially, but instead opted to star in ''[[Eyes Wide Shut]]''.&lt;ref&gt;{{cite web|last=Vejvoda|first=Jim|url=https://www.ign.com/articles/2001/08/03/vampire-director-named-prime-suspect|title=''Vampire'' Director Named Prime ''Suspect''|date=August 3, 2001|website=IGN|accessdate=May 28, 2022}}&lt;/ref&gt; Universal then courted [[Ben Affleck]] to rewrite Penn's script, but Affleck would depart the project over creative differences in November 1999.&lt;ref&gt;{{cite web|last=Green|first=Willow|url=https://www.empireonline.com/movies/news/suspect-zero-stalls/|title=''Suspect Zero'' Stalls|date=November 14, 1999|website=Empire Online|accessdate=May 28, 2022}}&lt;/ref&gt; [[Christian Duguay (director)|Christian Duguay]] would sign on to direct the film, while [[Paul Schrader]] and [[Richard Friedenberg]] were brought in for subsequent script revisions.&lt;ref&gt;{{cite web|last=Vejvoda|first=Jim|url=https://www.ign.com/articles/2001/08/03/vampire-director-named-prime-suspect|title=''Vampire'' Director Named Prime ''Suspect''|date=August 3, 2001|website=IGN|accessdate=May 28, 2022}}&lt;/ref&gt; By August 2001, [[E. Elias Merhige]] replaced Duguay as director, with production gearing up to start in fall of that year.&lt;ref&gt;{{cite web|last=Vejvoda|first=Jim|url=https://www.ign.com/articles/2001/08/03/vampire-director-named-prime-suspect|title=''Vampire'' Director Named Prime ''Suspect''|date=August 3, 2001|website=IGN|accessdate=May 28, 2022}}&lt;/ref&gt;

645 * [[Sylvester Stallone]]&lt;ref name=&quot;msn.com&quot;/&gt;

646 * [[Sylvester Stallone]]

647 '''''Daylight''''' is a 1996 American [[disaster film]] directed by [[Rob Cohen]] and starring [[Sylvester Stallone]], [[Amy Brenneman]], [[Viggo Mortensen]], [[Dan Hedaya]], [[Stan Shaw]], [[Jay O. Sanders]], [[Karen Young (actress)|Karen Young]] and [[Danielle Harris]]. Its plot concerns a group of people attempting to find their way out of an underground tunnel in [[Manhattan]], after explosions cause both ends of the tunnel to collapse.

648 * [[Sylvester Stallone]] as Chief Kit Latura, a disgraced former New York City Emergency Medical Services (EMS) chief who tries to save the survivors trapped inside the tunnel. Rob Cohen originally wanted [[Nicolas Cage]] to play Kit Latura, but Universal execs felt he was more of a &quot;character actor&quot;, and Stallone was more commercially viable

649 About the film, Stallone simply said: &quot;The premise was really good, but it didn't deliver&quot;.&lt;ref&gt;{{Cite web|url=https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/|title=Sylvester Stallone Feels Robbed of an Ownership Stake in 'Rocky': 'I Was Furious'|first1=Claudia|last1=Eller|date=July 23, 2019}}&lt;/ref&gt;

650 The film won a [[Golden Reel Award (Motion Picture Sound Editors)|Golden Reel Award]] for Best [[Sound editor (filmmaking)|Sound Editing]] and was nominated for an [[Academy Award]] in the same category ([[Richard L. Anderson (sound effects editor)|Richard L. Anderson]] and [[David A. Whittaker]]). It was also nominated for two [[Golden Raspberry Awards]], [[Golden Raspberry Award for Worst Actor|Worst Actor]] (Sylvester Stallone) and [[Golden Raspberry Award for Worst Original Song|Worst Original Song]] (''Whenever There Is Love''). For the 1996 [[Stinkers Bad Movie Awards]], Stallone was nominated for Worst Actor too.

651 The movie was famous for introducing the [[Panerai]] special line of watches (Panerai Daylight) by the request of Stallone himself.&lt;ref&gt;{{Cite web|url=https://paneraireview.com/panerai-history/panerai-radiomir-hidden-history-5-well-kept-facts/|title=Panerai Radiomir hidden history - 5 well kept facts|date=December 18, 2016}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|title=PAM 225 SLYTECH: A watch fit for Rambo {{!}} WatchBox|url=https://www.thewatchbox.com/blog/pam-225-slytech-watch-fit-rambo.html|access-date=July 11, 2021|website=www.thewatchbox.com|language=en}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|title=Sylvester Stallone's Watch Auction Has Serious Hollywood Pedigree|website=[[Bloomberg News]]|date=November 13, 2020 |url=https://www.bloomberg.com/news/articles/2020-11-13/sylvester-stallone-phillips-watch-auction-daylight-panerai-richard-mille|url-status=live|archive-url=https://web.archive.org/web/20201113082915/https://www.bloomberg.com/news/articles/2020-11-13/sylvester-stallone-phillips-watch-auction-daylight-panerai-richard-mille |archive-date=November 13, 2020 }}&lt;/ref&gt;

652 Located at the confluence of the Caney Fork, the [[Collins River]] and the [[Rocky River (Tennessee)|Rocky River]], is Great Falls Lake. This [[reservoir (water)|reservoir]] is impounded by [[Great Falls Dam (Tennessee)|Great Falls Dam]], a project of the former [[Tennessee Electric Power Company]], now owned and operated by the [[Tennessee Valley Authority]] (TVA). This is the only dam outside of the [[Tennessee River]] drainage system directly operated by TVA. This dam impounds a very small but very deep lake due to the depth of the gorges carved by the rivers it impounds. This area was something of a resort area in the early 20th century when such projects were uncommon, especially in the southeastern [[United States]], but other than a few cabins, there is little evidence of this today, as the area has been largely supplanted by larger, more modern developments. The dam is named for the Great Falls of the Caney Fork, caused by the descent of the stream off of the [[Highland Rim]] to the level of the [[Nashville Basin]]. Located on the lake is [[Rock Island State Park (Tennessee)|Rock Island State Park]], developed on the site of former woolen mills in the 19th century predating the electrical development. This area was used for a considerable number of exterior shots and stunts in the [[Sylvester Stallone]] film, ''[[The Specialist]]''.

653 * [[Sylvester Stallone]] improvised an ad-hoc poncho from discarded industrial grade canvas as [[John Rambo]] on the set of the 1982 movie ''[[First Blood]]''. Behind the scenes when filming, Stallone found the leftover canvas underneath an abandoned truck and used his signature survival knife to make a hole in the top fold and turned it into the poncho worn by the Rambo character and describes it having saved him from the freezing cold bitter rain of Northern Canada.&lt;ref&gt;{{cite web | url=https://www.menshealth.com/entertainment/a20951712/sylvester-stallone-instagram-rambo-costume/ | title=Sylvester Stallone Just Revealed the Badass Story Behind His 'Rambo' Outfit | date=29 May 2018 }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://entertainment.ha.com/itm/movie-tv-memorabilia/costumes/-john-rambo-s-poncho-from-first-blood-/a/7111-89049.s | title=&quot;John Rambo's&quot; Poncho from &quot;First Blood.&quot;... Movie/TV Memorabilia &amp;#124; Lot #89049 }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://www.julienslive.com/lot-details/index/catalog/400/lot/168196/SYLVESTER-STALLONE-JOHN-RAMBO-SCREEN-MATCHED-CANVAS-PONCHO-FROM-RAMBO-FIRST-BLOOD-WITH-DVD | title=Sylvester Stallone John Rambo Screen-Matched Canvas Poncho from Rambo: First Blood (With DVD) }}&lt;/ref&gt;

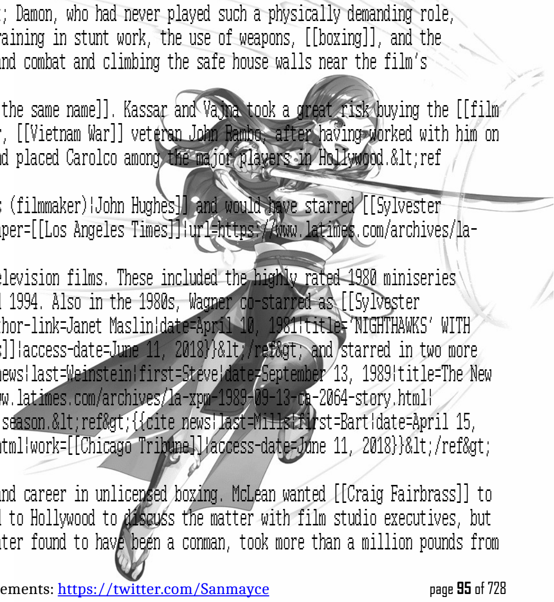
654 Liman approached a wide range of actors for the role of Bourne, including [[Brad Pitt]],&lt;ref name=Pitt/&gt; who turned it down to star in ''[[Spy Game]]'',&lt;ref&gt;{{cite news|url=https://variety.com/2000/film/news/inside-moves-187-1117822077/|title=Inside Moves|work=Variety|date=May 25, 2000|access-date=May 25, 2015}}&lt;/ref&gt; as well as [[Russell Crowe]], [[Arnold Schwarzenegger]], [[Tom Cruise]] and [[Sylvester Stallone]], before he eventually cast Damon. Liman found that Damon understood and appreciated that, though ''The Bourne Identity'' would have its share of action, the focus was primarily on character and plot.&lt;ref name=&quot;BBC&quot;&gt;{{cite web |last=Hanrahan |first=Denise |work=BBC.co.uk |title=Interview with Doug Liman | url=https://www.bbc.co.uk/films/2002/09/05/doug_liman_the_bourne_identity_interview.shtml |access-date=March 14, 2007}}&lt;/ref&gt; Damon, who had never played such a physically demanding role, insisted on performing many of the stunts himself. With stunt choreographer Nick Powell, he underwent three months of extensive training in stunt work, the use of weapons, [[boxing]], and the Filipino martial art [[eskrima]]. He eventually performed a significant number of the film's stunts himself, including hand-to-hand combat and climbing the safe house walls near the film's conclusion.&lt;ref name=&quot;DVDDocumentary&quot; /&gt;

655 Carolco's first major success was ''[[First Blood]]'' (1982), an adaptation of [[David Morrell]]'s [[First Blood (novel)|novel of the same name]]. Kassar and Vajna took a great risk buying the [[film rights]] to the novel (for $385,000) and used the help of European bank loans to cast [[Sylvester Stallone]] as the lead character, [[Vietnam War]] veteran John Rambo, after having worked with him on the [[John Huston]] film ''[[Escape to Victory]]'' (1981). The risk paid off after ''First Blood'' made $120 million worldwide, and placed Carolco among the major players in Hollywood.&lt;ref name=&quot;prince&quot;&gt;Prince, p. 144.&lt;/ref&gt;

656 Carolco also attempted to make ''Bartholomew vs. Neff'', a comedy film that was to have been written and directed by [[John Hughes (filmmaker)|John Hughes]] and would have starred [[Sylvester Stallone]] and [[John Candy]].&lt;ref&gt;{{cite news|title=SHORT TAKES : Stallone in Line for Comedy Role|date=30 July 1990|newspaper=[[Los Angeles Times]]|url=https://www.latimes.com/archives/la-xpm-1990-07-30-ca-997-story.html|access-date=26 February 2015}}&lt;/ref&gt;

657 Following the cancellation of ''The Bionic Woman'' in 1978, Wagner continued to act, predominantly in television miniseries and television films. These included the highly rated 1980 miniseries ''[[Scruples (TV miniseries)|Scruples]]'', as well as three made-for-TV ''Bionic'' reunion movies with Lee Majors between 1987 and 1994. Also in the 1980s, Wagner co-starred as [[Sylvester Stallone]]'s ex-wife in his 1981 movie ''[[Nighthawks (1981 film)|Nighthawks]]''&lt;ref&gt;{{cite news|last=Maslin|first=Janet|author-link=Janet Maslin|date=April 10, 1981|title='NIGHTHAWKS' WITH SYLVESTER STALLONE|url=https://www.nytimes.com/1981/04/10/movies/nighthawks-with-sylvester-stallone.html|work=[[The New York Times]]|access-date=June 11, 2018}}&lt;/ref&gt; and starred in two more weekly television series, ''[[Jessie (1984 TV series)|Jessie]]'' (1984) and ''[[A Peaceable Kingdom]]'' (1989),&lt;ref&gt;{{cite news|last=Weinstein|first=Steve|date=September 13, 1989|title=The New TV Season : CBS' Animal Park : Network Spends Millions of Dollars on a Zoo for Lindsay Wagner's 'Peaceable Kingdom'|url=https://www.latimes.com/archives/la-xpm-1989-09-13-ca-2064-story.html|work=[[Los Angeles Times]]|access-date=June 11, 2018}}&lt;/ref&gt; though both were cancelled with neither completing their first season.&lt;ref&gt;{{cite news|last=Mills|first=Bart|date=April 15, 1990|title=RELATIONSHIPS GO TO 'HEART' OF WAGNER MINI|url=https://www.chicagotribune.com/news/ct-xpm-1990-04-15-9002050859-story.html|work=[[Chicago Tribune]]|access-date=June 11, 2018}}&lt;/ref&gt; In 1983, she also appeared in an episode of Lee Majors' series ''[[The Fall Guy]]''.

658 In his autobiography, McLean recounts that various [[film studio]]s had expressed an interest in making a film based on his life and career in unlicensed boxing. McLean wanted [[Craig Fairbrass]] to portray him as he had known the actor for some time, and considered Fairbrass resembled himself as a younger man. McLean travelled to Hollywood to discuss the matter with film studio executives, but their preference for [[Sylvester Stallone]] for the part caused McLean to discontinue negotiations. One supposed film promoter, later found to have been a conman, took more than a million pounds from McLean and disappeared. The plan fell into hiatus on McLean's death.{{fact|date=October 2023}}



659 In 2011, the metal Kawkawa Bridge was demolished; previously, it was featured in the 1982 [[Rambo (film series)|Rambo]] film, ''[[First Blood]]''.&lt;ref&gt;{{cite web |title=Hope, B.C. celebrates 'Rambo bridge' before demolition |url=https://bc.ctvnews.ca/hope-b-c-celebrates-rambo-bridge-before-demolition-1.668039 |website=CTV News |archive-url=https://web.archive.org/web/20200820180337/https://bc.ctvnews.ca/hope-b-c-celebrates-rambo-bridge-before-demolition-1.668039 |archive-date=August 20, 2020 |date=July 9, 2011 |url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Rambo-Tour&quot;&gt;{{cite web |title=Rambo First Blood Filming Location Self-Guided Tour |url=https://hopebc.ca/rambo-first-blood-filming-location-tour-self-guided |website=District of Hope |date=14 October 2019 |archive-url=https://web.archive.org/web/20201214010545/https://hopebc.ca/rambo-first-blood-filming-location-tour-self-guided |archive-date=December 14, 2020 |url-status=live}}&lt;/ref&gt; In 2020, a wood carved statue of [[Sylvester Stallone]] character [[John Rambo]] was erected in Hope.&lt;ref&gt;{{cite web |title=Sylvester Stallone 'very proud' of new Rambo carving in B.C. town |url=https://bc.ctvnews.ca/sylvester-stallone-very-proud-of-new-rambo-carving-in-b-c-town-1.5069785 |website=CTV News |archive-url=https://web.archive.org/web/20201214010240/https://bc.ctvnews.ca/sylvester-stallone-very-proud-of-new-rambo-carving-in-b-c-town-1.5069785 |archive-date=December 14, 2020 |date=August 18, 2020 |url-status=live}}&lt;/ref&gt;

660 Hope has been a popular [[Filming location|location]] to shoot films. ''[[First Blood]]'' (1982), the first [[Rambo (film series)|Rambo film]], starring [[Sylvester Stallone]], [[Brian Dennehy]], and [[Richard Crenna]], was filmed almost entirely in and around Hope,&lt;ref name=&quot;Rambo-Tour&quot;/&gt; as was ''[[Shoot to Kill (1988 film)|Shoot to Kill]]'' (1988), starring [[Sidney Poitier]], [[Tom Berenger]] and [[Kirstie Alley]]. ''[[K2 (film)|K2]]'' (1992) was also filmed nearby, with the area's mountains standing in for the [[Himalayas]].&lt;ref name=&quot;HVG-2008&quot; /&gt;{{RP|20}}

661 In 2022, Noujaim served as an executive producer on ''[[Flight/Risk]]'' directed by Amer and [[Omar Mullick]] for [[Amazon Studios]].&lt;ref&gt;{{cite web|url=https://www.enigma-mag.com/flight-risk/|title=Flight/Risk: The Film That Changed the Way We Look at Aircraft|website=eniGma Magazine|first=Rawya|last=Lamei|access-date=August 20, 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/|title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan', 'Good Night Oppy' &amp; More|website=Deadline Hollywood|first=Anthony|last=D'Alessandro|date=August 1, 2022|access-date=August 20, 2023}}&lt;/ref&gt; In 2023, Noujaim served as an executive producer on ''[[Defiant (2023 film)|Defiant]]'' directed by Amer, focusing on diplomacy and [[disinformation in the Russian invasion of Ukraine]].&lt;ref&gt;{{cite web|url=https://tiff.net/events/defiant|title=Defiant|website=[[2023 Toronto International Film Festival]]|access-date=September 12, 2023}}&lt;/ref&gt; Noujaim is directing and executive producing alongside [[Vikram Gandhi]], ''The Man Will Burn'', a documentary series revolving around [[Burning Man]].&lt;ref&gt;{{cite web|url=https://deadline.com/2023/09/burning-man-docuseries-in-the-works-rain-chaos-30-years-of-archive-1235540766/|title=Burning Man Docuseries In The Works; Project Includes Crazy Scenes Of Rain Chaos &amp; 30 Years Of Archive|website=Deadline Hollywood|first=Peter|last=White|date=September 8, 2023|access-date=September 12, 2023}}&lt;/ref&gt;

662 * Although it is not heard, the song is mentioned by name in the 1991 Sylvester Stallone movie, ''[[Oscar (1991 film)|Oscar]]''.

663 *{{cite web |url=https://zeenews.india.com/entertainment/movies/navratri-special-top-5-bollywood-films-which-are-female-oriented_1804864.html|title=Navratri special: Top 5 Bollywood films which are female oriented! |date=1 October 2015 |website=Zee News|language=en|access-date=25 June 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://www.deccanherald.com/entertainment/was-gadar-patriotic-heres-what-sunny-deol-thought-about-films-theme-2648583 |title=Was 'Gadar' patriotic? Here's what Sunny Deol thought about film's theme}}&lt;/ref&gt; Deol's portrayals personified [[masculinity]] and [[Classical radicalism|radicalism]] in India.&lt;ref&gt;{{cite news |url=https://timesofindia.indiatimes.com/entertainment/hindi/bollywood/photo-features/sunny-deol-films-that-pack-a-punch-literally/photostory/44871194.cms |title=Sunny Deol films that pack a punch, literally! |newspaper=The Times of India}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://www.thehansindia.com/cinema/sunny-deol-enjoyed-doing-patriotic-themes--518780 |title=Sunny Deol enjoyed doing 'patriotic themes' |date=6 April 2019}}&lt;/ref&gt; At the peak of his career, Deol was known for his muscular physique.&lt;ref&gt;{{cite news |url=https://timesofindia.indiatimes.com/entertainment/hindi/bollywood/news/gadar-2-actor-sunny-deol-has-these-words-of-wisdom-for-budding-actors/articleshow/102694655.cms?from=mdr |title='Gadar 2' actor Sunny Deol has these words of wisdom for budding actors |newspaper=The Times of India |date=13 August 2023}}&lt;/ref&gt; He has been noted as one of the pioneers who kickstarted the trend of bodybuilding in Bollywood.&lt;ref&gt;{{Cite news |title=सलमान खान नहीं.... तो कौन है बॉलीवुड का पहला बॉडी बिल्डर? जानकर लगेगा तगड़ा झटका | url=https://hindi.news18.com/news/entertainment/bollywood-neither-salman-khan-nor-sanjay-dutt-gavin-packard-was-first-body-builder-of-bollywood-know-about-forgotten-villain-of-industry-6194877.html | work=News18 हिंदी |language=hi |date=15 May 2023 |access-date=13 May 2024}}&lt;/ref&gt; Deol was one of the prominent actors of Hindi cinema in the 1980s, 1990s, and early 2000s, appearing in [[Box Office India]]'s list of top actors nine times from 1989 to 2002. Deol topped the list twice, in 1997 and 2001.&lt;ref&gt;{{Cite web |date=29 October 2013 |title=Boxofficeindia.com |url=http://boxofficeindia.com/cpages.php?pageName=top_actors |access-date=9 January 2024 |archive-url=https://web.archive.org/web/20131029191201/http://boxofficeindia.com/cpages.php?pageName=top_actors |archive-date=29 October 2013 }}&lt;/ref&gt; Deol was one of the highest paid actors of Hindi cinema during the 1990s.&lt;ref&gt;{{Cite web |title=90s में सबसे ज्यादा फीस लेते थे ये 9 स्टार, चौंका देगा नंबर 1 का नाम |url=https://hindi.asianetnews.com/webstories/entertainment/bollywood/sunny-deol-to-salman-khan-akshay-kumar-shah-rukh-khan-ajay-devgn-meet-90s-highest-paid-actors-gga-qg72akh |access-date=9 January 2024 |website=hindi.asianetnews.com |language=hi}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |title=बॉलीवुड: ये हैं 90 के दशक के सबसे महंगे सितारे, इन्हें फिल्म में लेने वाले निर्माताओं की खाली हो जाती थी जेब |url=https://www.amarujala.com/photo-gallery/entertainment/bollywood/shahrukh-khan-to-ajay-devgn-meet-the-highest-paid-actors-of-bollywood-from-90s |access-date=9 January 2024 |website=Amar Ujala |language=hi}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |date=12 May 2023 |title=Sunny Deol To Salman Khan, 5 Highest Paid Actors Of The 90s |url=https://www.news18.com/entertainment/bollywood/sunny-deol-to-salman-khan-5-highest-paid-actors-of-the-90s-7798609.html |access-date=9 January 2024 |website=News18 |language=en}}&lt;/ref&gt; Since 1990, Deol has the third most bumper openings at the box office with a total of 10, behind only [[Shah Rukh Khan]] (18) and [[Salman Khan]] (14).&lt;ref&gt;{{Cite web |title=Bharat Is Fourteenth Bumper Opening For Salman Khan - Box Office India | url=https://www.boxofficeindia.com/report-details.php?articleid=5076 |access-date=28 March 2024 |website=www.boxofficeindia.com}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |date=8 September 2023 |title=Box Office: Shah Rukh Khan delivers the 18th bumper opener of his career with Jawan; Most since the year 1990 |url=https://www.pinkvilla.com/entertainment/box-office/box-office-shah-rukh-khan-delivers-the-18th-bumper-opener-of-his-career-with-jawan-most-since-the-year-1990-1241735 |access-date=28 March 2024 |website=PINKVILLA |language=en}}&lt;/ref&gt; He has also given three record openers since 1990 including ''[[Kshatriya (film)|Kshatriya]]'' (1993), ''[[Border (1997 film)|Border]]'' (1997), and ''[[Indian (2001 film)|Indian]]'' (2001).&lt;ref name=&quot;auto1&quot;/&gt; Deol is the first and only Indian actor to deliver an all-time blockbuster as a lead hero after turning 60, he was 66 when he accomplished this landmark with his 2023 release ''Gadar 2''.&lt;ref&gt;{{Cite news | last=Singh |first=Simran |date=6 April 2024 |title=Meet star with 30 flops, no solo hit in 22 years, career got ruined; then gave highest-grossing film at age 66, now he.. |url=https://www.dnaindia.com/bollywood/report-meet-star-with-30-flops-sunny-deol-no-solo-hit-in-22-years-career-got-ruined-then-highest-grossing-film-at-66-3084324/amp |work=DNA |access-date=8 April 2024}}&lt;/ref&gt; He has been compared to [[Sylvester Stallone]] that earned him the moniker of &quot;Indian Rambo&quot;.&lt;ref&gt;{{Cite web |last=Pal |first=Deepanjana |date=1 October 2023 | title=Angry Young Man to Grumpy Old Man: How Sunny Deol Expanded the Definition of Hero |url=https://www.filmcompanion.in/features/bollywood-features/angry-young-man-to-grumpy-old-man-how-sunny-deol-expanded-the-definition-of-hero |website=Filmcompanion |access-date=8 April 2024}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Bhattacharya |first=Roshmila |date=25 March 2015 |title=BACKSTAGE PASS - Sunny is like an Indian Sylvester Stallone |url=https://timesofindia.indiatimes.com/entertainment/hindi/bollywood/news/backstage-pass-sunny-is-like-an-indian-sylvester-stallone/articleshow/46688471.cms |work=The Times of India |access-date=8 April 2024 |issn=0971-8257}}&lt;/ref&gt;

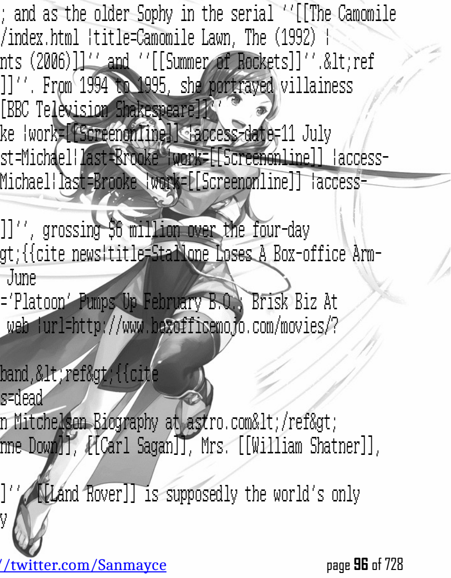
664 Prochnow dubbed [[Sylvester Stallone]]'s voice in the German version of ''[[Rocky]]'' (1976) and ''[[Rocky II]]'' (1979), as well as ''[[F.I.S.T. (film)|F.I.S.T]]'' and ''[[Paradise Alley]]'' (both 1978). He later acted alongside Stallone in the 1995 movie ''[[Judge Dredd (film)|Judge Dredd]]''. After the retirement of Stallone's long-time voice actor [[Thomas Danneberg]], Prochnow assumed this job in 2018 with ''[[Creed II]]''. He usually also dubs his own roles in English-language productions.&lt;ref&gt;{{cite web |url=https://www.synchronkartei.de/sprecher/525/2 |title=Deutsche Synchronkartei (german) |access-date=1 June 2021 |work=synchrondatei.de}}&lt;/ref&gt;

665 She also appeared in the [[Woody Allen]] films ''[[Crimes and Misdemeanors]]'' (1989) and ''[[Mighty Aphrodite]]'' (1995).&lt;ref name=&quot;franco&quot; /&gt; She appeared in the [[Sylvester Stallone]] film ''[[Daylight (1996 film)|Daylight]]'' (1996).&lt;ref&gt;{{cite news| url=https://www.nytimes.com/1996/12/15/movies/daylight.html | work=[[New York Times]] | title=Daylight | first=Peter M. | last=Nichols | date=December 15, 1996 | access-date=July 10, 2023}}&lt;/ref&gt; Later appearances in films included her portrayal of [[Mary of Teck|Queen Mary]] in the 2010 Oscar winning British film ''[[The King's Speech]]'' and her portrayal of Eva Rose opposite Jerry Lewis in the 2016 film ''[[Max Rose (film)|Max Rose]]''.&lt;ref name=&quot;Claire Bloom&quot; /&gt;&lt;ref name=&quot;KF&quot;&gt;{{cite news|last=Fujishima|first=Kenji|url=https://blogs.wsj.com/speakeasy/2011/05/25/charlie-chaplin-is-the-father-claire-bloom-never-had/|title=Charlie Chaplin Is the Father Claire Bloom Never Had|date=25 May 2011|newspaper=[[The Wall Street Journal]]|department=Speakeasy|access-date=28 May 2015}}&lt;/ref&gt; In television she acted in ''[[The Mirror Crack'd from Side to Side|The Mirror Crack'd]]'', the last of the BBC [[Miss Marple (TV series)|Miss Marple]] adaptations in 1992;&lt;ref&gt;{{cite news| url=https://www.independent.co.uk/news/uk/swansong-for-miss-marple-1560939.html | work=[[The Independent]] | title=Swan Song for Miss Marple? | date=2 December 1992 | access-date=11 July 2023}}&lt;/ref&gt; and as the older Sophy in the serial ''[[The Camomile Lawn (TV serial)|The Camomile Lawn]]'' (1992) on Britain's [[Channel 4]].&lt;ref&gt;{{cite web |url=http://www.screenonline.org.uk/tv/id/1295276/index.html |title=Camomile Lawn, The (1992) | first=Sergio|last=Angelini |work=[[Screenonline]] |access-date=11 July 2023}}&lt;/ref&gt; Recent mini-series work includes ''[[The Ten Commandments (2006)]]'' and ''[[Summer of Rockets]]''.&lt;ref name=&quot;Claire Bloom&quot; /&gt; On continuing television series, she has appeared on the New York-based ''[[Law &amp; Order: Criminal Intent]]''. From 1994 to 1995, she portrayed villainess Orlena Grimaldi on the daytime drama ''[[As the World Turns]]''.&lt;ref name=&quot;KF&quot; /&gt; She also had major roles in several of the ''[[BBC Television Shakespeare]]'' presentations&lt;ref&gt;{{cite web |url=http://www.screenonline.org.uk/tv/id/526970/index.html |title=Henry VIII (1979) |first=Michael|last=Brooke |work=[[Screenonline]] |access-date=11 July 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://www.screenonline.org.uk/tv/id/527101/index.html |title=Hamlet, Prince of Denmark (1980) |first=Michael|last=Brooke |work=[[Screenonline]] |access-date=11 July 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://www.screenonline.org.uk/tv/id/527675/index.html |title=Cymbeline (1983) |first=Michael|last=Brooke |work=[[Screenonline]] |access-date=11 July 2023}}&lt;/ref&gt; and has led workshops on Shakespearean performance practices.&lt;ref name=&quot;franco&quot; /&gt;

666 The film debuted at number three at the US box office behind ''[[Platoon (film)|Platoon]]'' and ''[[Outrageous Fortune (film)|Outrageous Fortune]]'', grossing $6 million over the four-day [[President's Day]] weekend, surpassing the other opener, ''[[Over the Top (1987 film)|Over the Top]]'' starring [[Sylvester Stallone]].&lt;ref&gt;{{cite news|title=Stallone Loses A Box-office Arm-wrestle|work=[[Los Angeles Times]]|date=19 February 1987|url=https://www.latimes.com/archives/la-xpm-1987-02-19-ca-4353-story.html|access-date=2 June 2012|first=Jack|last=Mathews}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine|magazine=[[Variety (magazine)|Variety]]|date=February 18, 1987|page=3|title='Platoon' Pumps Up February B.O.; Brisk Biz At Top}}&lt;/ref&gt; To date, the film has grossed a total of $42.7 million in the United States and Canada.&lt;ref name=&quot;mojo&quot;&gt;{{cite web |url=http://www.boxofficemojo.com/movies/?id=mannequin.htm |title=''Mannequin'' (1987) |access-date=14 May 2013 }}&lt;/ref&gt;

667 Mitchelson's celebrity clients included [[Pamela Mason]] (wife of [[James Mason]]), who received a $2 million divorce settlement from her ex-husband,&lt;ref&gt;{{cite web|url=http://www.astro.com/astro-databank/Mitchelson_Marvin |title=Mitchelson Marvin - Astro-Databank |accessdate=February 17, 2010 |url-status=dead |archiveurl=https://web.archive.org/web/20110728062143/http://www.astro.com/astro-databank/Mitchelson_Marvin |archivedate=July 28, 2011 }} Marvin Mitchelson Biography at astro.com&lt;/ref&gt; [[Robert De Niro]], [[Mickey Rooney]], [[Sylvester Stallone]], [[Zsa Zsa Gabor]], [[Joan Collins]], [[Mel Tormé]], [[Bianca Jagger]], [[Lesley-Anne Down]], [[Carl Sagan]], Mrs. [[William Shatner]], and many ex-wives of errant playboy sheiks.

668 Thirty-one 101s were converted by Land Rover with styled bodyshells for the 1995 [[Sylvester Stallone]] film ''[[Judge Dredd (film)|Judge Dredd]]''. [[Land Rover]] is supposedly the world's only surviving vehicle manufacturer in 2139 when the film is set, and the green oval logo can be seen on the side of the vehicles, known as &quot;City CABs&quot;.&lt;ref&gt;[http://www.talkingpix.co.uk/Reviews%20-%20Judge%20Dredd.html Talkingpix.com : Judge Dredd]&lt;/ref&gt;



669 Also on the DVD, the film's writer and director [[Sylvester Stallone]] wanted to cast a real boxer in the role of Dixon, as he thought it would be easier to teach a boxer how to act than to teach an actor how to box convincingly. Tarver is one of only three fighters to actually defeat &quot;Rocky Balboa&quot; in film.

670 It was launched in [[New York City]] on October 22, 1991, with the backing of Hollywood stars [[Sylvester Stallone]], [[Arnold Schwarzenegger]] and [[Bruce Willis]]. The actors recruited were paid for their appearances and endorsements through an [[employee stock ownership plan]].&lt;ref&gt;{{Cite news |url=https://economictimes.indiatimes.com/stars-like-bruce-willis-arnold-schwarzenegger-and-sylvester-stallone-endorse-planet-hollywood/articleshow/9937771.cms |title=Stars like Bruce Willis, Arnold Schwarzenegger and Sylvester Stallone endorse Planet Hollywood |date=September 11, 2011 |work=The Economic Times |access-date=June 22, 2018 |publisher=Bennett, Coleman &amp; Co. Ltd.}}&lt;/ref&gt; Further celebrity endorsement included actors [[Demi Moore]], [[Whoopi Goldberg]], [[Jean-Claude Van Damme]], [[Don Johnson]], [[Cindy Crawford]], [[Melanie Griffith]], [[Tom Arnold (actor)|Tom Arnold]], [[Wesley Snipes]], and [[Danny Glover]]; director [[John Hughes (filmmaker)|John Hughes]]; comedian [[Roseanne Barr]]; and [[The Beatles]] drummer [[Ringo Starr]].

671 Planet Hollywood was opened in late 1991 in [[New York City]],&lt;ref name=&quot;idch&quot; /&gt;&lt;ref&gt;{{Cite news|date=September 6, 1991|title=Concierges: Baedekers In the Flesh; How to Get in Touch With the Highly Recommended|language=en-US|work=The New York Times|url=https://www.nytimes.com/1991/09/06/arts/concierges-baedekers-flesh-get-touch-with-highly-recommended.html|access-date=November 19, 2020|issn=0362-4331}}&lt;/ref&gt; at the base of two adjacent office buildings at [[130 West 57th Street|130]] and [[140 West 57th Street]].&lt;ref&gt;{{Cite news|last=Gray|first=Christopher|date=May 9, 1999|title=Streetscapes /57th Street Between Avenue of the Americas and Seventh Avenue; High and Low Notes of a Block With a Musical Bent|language=en-US|work=The New York Times|url=https://www.nytimes.com/1999/05/09/realestate/streetscapes-57th-street-between-avenue-americas-seventh-avenue-high-low-notes.html|access-date=November 18, 2020|issn=0362-4331}}&lt;/ref&gt; [[Arnold Schwarzenegger]], [[Sylvester Stallone]] and [[Bruce Willis]] were some of the actors who became investors/promoters.&lt;ref name=&quot;idch&quot; /&gt; By the next year, three more locations were opened: [[London]], southern California, and [[Chicago]]. In 1993, the company opened two new locations, Washington, D.C., and Cancun, Mexico, while arranging leases for five new locations. Each new unit, opening with a gala event, generated nearly $15 million revenue in their first operational year.&lt;ref name=&quot;idch&quot; /&gt;

672 [[Category:Sylvester Stallone]]

673 * [[Sage Stallone]] (1976-2012), American musician, actor and film producer/director; son of Sylvester Stallone

674 * [[Sistine Stallone]] (born 1998), American model and actress; daughter of Sylvester Stallone

675 * [[Sylvester Stallone]] (born 1946), American actor, screenwriter and film producer/director; son of Frank and Jackie Stallone

676 | {{marriage|[[Sylvester Stallone]]|1985|1987|end=divorced}}

677 '''Brigitte Nielsen''' ({{IPA|da|pʁiˈkitə ˈne̝lsn̩}}; born '''Gitte Nielsen'''; 15 July 1963) is a Danish actress, model, and singer. She began her career modelling for [[Greg Gorman]] and [[Helmut Newton]]. She subsequently acted in the 1985 films ''[[Red Sonja (1985 film)|Red Sonja]]'' and ''[[Rocky IV]]'', later returning to the [[Rocky (franchise)|''Rocky'' series]] in ''[[Creed II]]'' (2018). Nielsen married [[Sylvester Stallone]], with whom she starred in the 1986 film ''[[Cobra (1986 film)|Cobra]]''. She played a villain in ''[[Beverly Hills Cop II]]'' (1987) and starred as the Black Witch in the 1990s Italian film series ''[[Fantaghirò series|Fantaghirò]].'' She later built a career starring in [[B movie|B-movie]]s, hosting TV shows, and appearing on reality shows.

678 In 1985, Nielsen began her acting career in the fantasy film ''[[Red Sonja (1985 film)|Red Sonja]]'' alongside [[Arnold Schwarzenegger]]. That same year, she married [[Sylvester Stallone]]; the couple acted in two films together (''[[Rocky IV]]'' and ''[[Cobra (1986 film)|Cobra]]'') before divorcing in 1987.

679 [[File:Sylvester Stallone Brigitte Nielsen (cropped).jpg|thumb|Nielsen with [[Sylvester Stallone]], [[Ronald Reagan|Ronald]] and [[Nancy Reagan]] at the [[White House]], 1985]]

680 Nielsen became engaged to actor [[Sylvester Stallone]] during the production of ''[[Rocky IV]].'' They married after the release of the film, on 15 December 1985, at the [[Beverly Hills, California]] home of producer [[Irwin Winkler]].&lt;ref&gt;{{cite magazine|url=https://ew.com/article/1997/12/12/stallone-married-brigitte-12-years-ago/|title=A Rocky Marriage|magazine=[[Entertainment Weekly]]|date=12 December 1997|access-date=2011-12-24|archive-date=29 October 2014|archive-url=https://web.archive.org/web/20141029105514/http://www.ew.com/ew/article/0,,290699,00.html|url-status=live}}&lt;/ref&gt; They eventually divorced.&lt;ref&gt;{{cite web|url=https://people.com/archive/cover-story-gitte-finds-her-mark-vol-29-no-13/|date=4 April 1988|work=[[People (magazine)|People]]|title=Gitte Finds Her Mark|first1=Joanne |last1=Kaufman|first2= Jorgen |last2=Salomonsen|first3= Alan|last3= Richman|access-date=25 February 2019}}&lt;/ref&gt; Their marriage and divorce were both highly publicized by the tabloid press due to their involvement on ''Rocky IV'' and ''[[Cobra (1986 film)|Cobra]]''.&lt;ref name=&quot;autogenerated27&quot;&gt;Susan Zannos, ''Male Fitness Stars of TV and the Movies: Featuring Profiles of Sylvester Stallone, John Travolta, Bruce Willis and Wesley Snipes'', Mitchell Lane Publishers, 2000, page 27&lt;/ref&gt;&lt;ref&gt;[https://news.google.com/newspapers?nid=1755&amp;dat=19870723&amp;id=vqMcAAAAIBAJ&amp;sjid=RXAEAAAAIBAJ&amp;pg=7140,3399728 &quot;Stallone divorce stops Tabloid presses&quot;], ''[[Sarasota Herald-Tribune]]'' - 23 July 1987.&lt;/ref&gt;&lt;ref name=&quot;nytimes1997&quot;&gt;{{cite news|url=https://www.nytimes.com/1997/08/10/movies/stallone-seeks-a-serious-turn-for-the-better.html|title=Stallone Seeks a Serious Turn for the Better - Page 3 - ''The New York Times''|date=1997-08-10|access-date=2011-10-26|work=[[The New York Times]]}}&lt;/ref&gt;

681 On 9 July 2007, Nielsen checked herself into the Cri-Help [[Drug rehabilitation|rehabilitation]] facility in [[North Hollywood]] following what some reports claim was a family [[Intervention (counseling)|intervention]].&lt;ref&gt;{{cite web|url=http://www.contactmusic.com/news.nsf/article/nielsen%20in%20rehab_1037977 |title=Sylvester Stallone &amp;#124; Nielsen In Rehab |publisher=Contactmusic |access-date=2011-10-26}}&lt;/ref&gt; Her manager, Steven Tempone, confirmed on 19 July 2007 that she had checked into a rehabilitation center and said, &quot;All I know is it's something she did of her own free will and we're proud of her and wish her very well{{nbsp}}... When she gets out we'll have a big birthday party, and [[Coca-Cola]] only.&quot;{{cn|date=September 2023}}&lt;!-- &lt;ref&gt;{{Cite web|url=https://www.voanews.com/english/2007-07-20-voa40.cfm|archive-url=https://web.archive.org/web/20211210010511/https://www.voanews.com/english/2007-07-20-voa40.cfm|url-status=dead|archive-date=2021-12-10|title=Page doesn't exist|date=10 December 2021}}&lt;/ref&gt; --&gt; As of 22 July 2007, Nielsen was out of rehabilitation to attend the [[Comedy Central Roast of Flavor Flav|''Comedy Central Roast'' of Flavor Flav]]. She remarked that she felt &quot;like a new-born person&amp;nbsp;... I made a choice about a new life. It's not been easy but it was definitely time.&quot; While the date that Nielsen checked into rehabilitation was reported, she &quot;had been in treatment for a few weeks.&quot;&lt;ref&gt;{{cite web|last=Rodriguez |first=Brenda |url=https://people.com/celebrity/brigitte-nielsen-i-feel-reborn-after-rehab/ |title=Brigitte Nielsen: I Feel Reborn After Rehab - Brigitte Nielsen|work=People|date=23 July 2007|access-date=25 February 2019}}&lt;/ref&gt; On 10 January 2008, [[VH1]] began airing the reality TV series ''[[Celebrity Rehab]]''. Nielsen appeared on the show for alcoholism.&lt;ref&gt;{{cite web|url=http://www.vh1.com/shows/dyn/celebrity_rehab_with_dr_drew/series.jhtml |archive-url=https://web.archive.org/web/20071208143001/http://www.vh1.com/shows/dyn/celebrity_rehab_with_dr_drew/series.jhtml |url-status=dead |archive-date=8 December 2007 |title=Celebrity Rehab 5 &amp;#124; Show Cast, Episodes, Guides, Trailers, Web Exclusives, Previews |publisher=VH1.com |access-date=2011-10-26}}&lt;/ref&gt; In a January 2010, ''[[TV Guide]]'' story on her success with sobriety, ''Celebrity Rehab'' producer John Irwin stated, &quot;She's sober, and she quit smoking.&quot;&lt;ref name=&quot;Rochlin&quot; /&gt;

682 [[Category:Family of Sylvester Stallone]]</text>

683 In 1982, Survivor's breakthrough arrived when actor [[Sylvester Stallone]] asked them to provide the theme song for his movie ''[[Rocky III]]''. Stallone had heard &quot;Poor Man's Son&quot; and wanted a song similar to it and to [[Queen (band)|Queen]]'s &quot;[[Another One Bites the Dust]]&quot;.&lt;ref name=&quot;Grdn&quot;&gt;{{cite news |last1=Jack |first1=Malcolm |title=Survivor: how we made Eye of the Tiger |url=https://www.theguardian.com/music/2020/jan/27/how-we-made-eye-of-the-tiger-rocky-iii-survivor-sylvester-stallone |access-date=20 September 2023 |work=The Guardian |date=27 January 2020}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |url= http://www.rhapsody.com/survivor |title=Survivor profile |website= Rhapsody.com| access-date=November 9, 2019}}&lt;/ref&gt; The band agreed to his request and soon came up with &quot;[[Eye of the Tiger]]&quot;.&lt;ref&gt;{{cite news |title=Why the band behind 'Eye of the Tiger' didn't want it to be Kim Davis's fight song |first= Justin Wm. | last= Moyer|url= https://www.washingtonpost.com/news/morning-mix/wp/2015/09/09/why-the-band-behind-eye-of-the-tiger-hates-its-use-as-kim-daviss-fight-song/| newspaper= [[The Washington Post]] | date= September 9, 2015 | access-date= }}&lt;/ref&gt;

684 ''The Beaver Trilogy'' combines three separate vignettes that were filmed at different times, in 1979, 1981, and 1984. The first, titled ''The Beaver Kid'', is a short documentary about the exploits of &quot;Groovin' Gary&quot;, a performer that filmmaker Harris happened upon while filming for a [[Salt Lake City, Utah]] news station. Harris was in the parking lot of his workplace, testing out a [[color|color video]] [[video camera]] that the station had just acquired, when he stumbled upon Gary taking photographs of their news [[helicopter]]. Gary immediately launched into a number of celebrity impressions, including [[John Wayne]] and [[Sylvester Stallone]].

685 | starring   = [[David Carradine]]&lt;br /&gt;[[Simone Griffeth]]&lt;br /&gt;[[Sylvester Stallone]]&lt;br /&gt;[[Louisa Moritz]]&lt;br /&gt;[[Don Steele]]

686 '''''Death Race 2000''''' is a 1975 American [[Dystopian fiction|dystopian]] [[Science fiction film|science-fiction]] [[action film]] directed by [[Paul Bartel]] and produced by [[Roger Corman]] for [[New World Pictures]].&lt;ref&gt;{{cite web |title=Death Race 2000 (1975) |url=https://www.allmovie.com/movie/death-race-2000-v12891 |access-date=November 16, 2021 |website=[[AllMovie]]}}&lt;/ref&gt; Set in a dystopian American society in the year 2000, the film centers on the murderous Transcontinental Road Race, in which participants score points by striking and killing pedestrians. [[David Carradine]] stars as &quot;Frankenstein,&quot; the leading champion of the race, who is targeted by an underground rebel movement seeking to abolish the race. The cast also features [[Simone Griffeth]], [[Sylvester Stallone]], [[Mary Woronov]], [[Martin Kove]], and [[Don Steele]].

687 * [[Sylvester Stallone]] as &quot;Machine Gun&quot; Joe Viterbo

688 [[Sylvester Stallone]] was cast after Corman saw his performance in ''[[The Lords of Flatbush]]'' (1974).&lt;ref name=&quot;:2&quot; /&gt; Up until his star-making role in ''[[Rocky]]'' the following year, ''Death Race 2000'' was the actor's highest-profile performance.&lt;ref name=&quot;:1&quot; /&gt; At Bartel's direction, Stallone rewrote much of his character's dialogue.

689 Due to the film's low-budget, many scenes were shot on public roadways.&lt;ref name=&quot;:2&quot; /&gt; David Carradine and Sylvester Stallone performed most of their own driving stunts.&lt;ref name=&quot;:2&quot; /&gt; According to Corman, the custom-built cars were not street legal, and the stunt drivers refused to drive them where they could potentially be apprehended by police.&lt;ref name=&quot;:2&quot; /&gt; Meanwhile, [[Mary Woronov]] did not know how to drive a car, so her close-ups were shot with her car towed behind a flatbed truck.&lt;ref name=&quot;:2&quot; /&gt;

690 The film was nominated for two [[Golden Raspberry Award]]s: Burt Reynolds was nominated for [[Golden Raspberry Award for Worst Actor|Worst Actor]] and Christopher Reeve for [[Golden Raspberry Award for Worst Supporting Actor|Worst Supporting Actor]]. However, they respectively &quot;lost&quot; to [[Sylvester Stallone]] for ''[[Rambo III]]'' and [[Dan Aykroyd]] for ''[[Caddyshack II]]''.

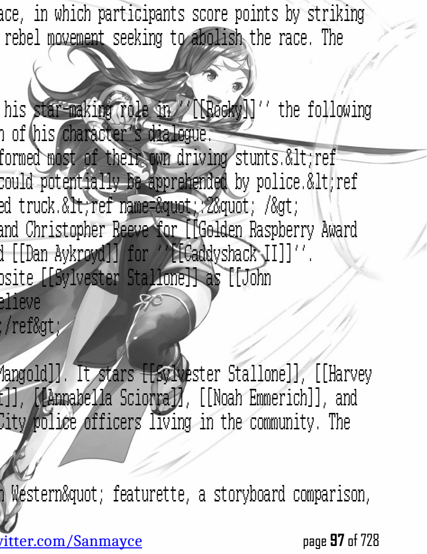
691 Dennehy was primarily known as a dramatic actor. His breakthrough role was as the overzealous sheriff Will Teasle in ''[[First Blood]]'' (1982) opposite [[Sylvester Stallone]] as [[John Rambo]].&lt;ref&gt;{{cite news |last=Canby |first=Vincent |author-link=Vincent Canby |date=November 22, 1976 |title=Film: 'Rocky,' Pure 30's Make-Believe |url=https://www.nytimes.com/1976/11/22/archives/film-rocky-pure-30s-makebelieve.html |work=The New York Times |access-date=September 23, 2016}}&lt;/ref&gt;

692 * [[Sylvester Stallone]]

693 '''''Cop Land''''' is a 1997 American [[neo-noir]] [[Crime film|crime]] [[Drama (film and television)|drama film]] written and directed by [[James Mangold]]. It stars [[Sylvester Stallone]], [[Harvey Keitel]], [[Ray Liotta]], and [[Robert De Niro]] as the main cast, with [[Peter Berg]], [[Janeane Garofalo]], [[Robert Patrick]], [[Michael Rapaport]], [[Annabella Sciorra]], [[Noah Emmerich]], and [[Cathy Moriarty]] in supporting roles. Stallone portrays the sheriff of a small New Jersey town who comes into conflict with the corrupt New York City police officers living in the community. The film received positive reviews and grossed $63.7 million on a $15 million budget.

694 * [[Sylvester Stallone]] as Sheriff Freddy Heflin

695 Extras include an audio commentary (with James Mangold, Sylvester Stallone, Robert Patrick, and producer Cathy Konrad), &quot;The Making of an Urban Western&quot; featurette, a storyboard comparison, two deleted scenes and the theatrical trailer.



696 The film posted solid box-office takings, got good reviews, and Stallone received positive critical notices for his performance. Yet Stallone has said the film was bad for his career. In a 2019 interview Stallone called Mangold &quot;the best director I ever worked with [but the film] actually worked in reverse. It was pretty good critically, but the fact that it didn't do a lot of box office, again it fomented the opinion that I had my moment and was going the way of the [[dodo bird]] and the [[Tasmanian tiger]].&quot;&lt;ref&gt;{{cite web |date= July 23, 2019 |last= Eller |first= Claudia |title= Sylvester Stallone Feels Robbed of an Ownership Stake in 'Rocky': 'I Was Furious' |url= https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/ | website= Variety |access-date= July 24, 2019 |archive-date= October 31, 2021 |archive-url= https://web.archive.org/web/20211031124140/https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/ |url-status= live }}&lt;/ref&gt;

697 | spouse   = {{marriage|[[Sylvester Stallone]]|May 17, 1997}}

698 '''Jennifer Lee Flavin Stallone''' ({{nee}} '''Flavin'''; born August 14, 1968) is an American model and entrepreneur. She is married to [[Sylvester Stallone]].

699 She graduated from [[El Camino Real High School]] in [[Woodland Hills, Los Angeles, California|Woodland Hills]], another district of Los Angeles; and then she began modeling with the representation of [[Elite Modeling Agency]] at the age of 19.&lt;ref name=&quot;haute&quot;&gt;{{Cite web|url=https://hauteliving.com/|title=Luxury Hotels, Travel, Dining, Lifestyle|website=Haute Living}}&lt;/ref&gt; She went on to appear as herself in reality shows, including ''[[Good Day Live]]'', ''[[American Gladiators (1989 TV series)|American Gladiators]]'', and ''[[The Contender (TV series)|The Contender]]'' - a [[boxing]] [[reality television]] series that also features [[Sylvester Stallone|Stallone]]. She also made a brief appearance in the 1990 film ''[[Rocky V]]''.

700 In 1988, Flavin met [[Sylvester Stallone]] in California at a [[Beverly Hills]] restaurant. They were in a relationship until 1994, when Stallone wrote her a letter saying that he was having an affair with model [[Janice Dickinson]], and that he was the father of her infant daughter.&lt;ref&gt;{{Cite web |url=https://people.com/archive/rocky-ending-vol-41-no-16/ |title=Rocky Ending | last=Schneider |first=Karen S.|date=March 2, 1994|website=People}}&lt;/ref&gt; However, after DNA tests had shown that he was not the father, the relationship ended.&lt;ref&gt;{{Cite web|url=https://people.com/archive/yo-angie-marry-me-vol-43-no-16/|title=Yo, Angie, Marry Me!|last=Levitt|first=Shelley|date=April 24, 1995|website=People}}&lt;/ref&gt;

701 On August 19, 2022, Flavin filed for a divorce from Stallone in the state of Florida.&lt;ref&gt;{{Cite web |title=Sylvester Stallone's Wife Jennifer Flavin Files for Divorce After 25 Years | url=https://www.tmz.com/2022/08/24/sylvester-stallone-wife-files-divorce-jennifer-flavin-money/ |access-date=August 24, 2022 |website=TMZ |date=August 24, 2022 |language=en}}&lt;/ref&gt; However, on September 23, Flavin and Stallone announced that they had reconciled.&lt;ref&gt;{{Cite web |title=Sylvester Stallone and Wife Jennifer Flavin Reconcile 1 Month After She Filed for Divorce | url=https://people.com/movies/sylvester-stallone-wife-jennifer-flavin-reconcile-after-divorce-filing/ |access-date=2023-09-30 |website=Peoplemag |language=en}}&lt;/ref&gt;

702 [[Category:Family of Sylvester Stallone]]</text>

703 *He is portrayed by [[Sylvester Stallone]] in the 1975 film ''[[Capone (1975 film)|Capone]]''. Nitti starts off as a bodyguard, assassin, and adviser under Capone ([[Ben Gazzara]]) before secretly betraying him and selling the IRS files that led to Capone's arrest for tax evasion. As the new head of the Chicago Outfit, he is last seen visiting the dying Capone at his [[Palm Island (Miami Beach)|Palm Island]] estate in 1946, a year before Capone's death and three years after Nitti's actual suicide.

704 *[[Sylvester Stallone]] (born 1946) - actor, director, screenwriter

705 At one point, the film was a serious [[action film]] starring [[Sylvester Stallone]],&lt;ref&gt;{{Cite news|url=https://www.cinemablend.com/news/1544540/why-sylvester-stallones-beverly-hills-cop-movie-never-happened|title=Why Sylvester Stallone's Beverly Hills Cop Movie Never Happened|date=2016-08-11|work=CINEMABLEND|access-date=2017-08-27}}&lt;/ref&gt; a project that Stallone later took with him and developed into [[Cobra (1986 film)|''Cobra'']]. Despite the troubled production history, the film was a massive critical and financial success, grossing over $300 million worldwide from a $15 million budget&lt;ref&gt;{{Cite web|url=https://www.boxofficemojo.com/movies/?id=beverlyhillscop.htm|title=Beverly Hills Cop (1984) - Box Office Mojo|website=www.boxofficemojo.com|language=en|access-date=2017-08-27}}&lt;/ref&gt; and earning rave reviews from critics like [[Janet Maslin]]&lt;ref&gt;{{Cite news|url=http://movies.nytimes.com/movie/review?res=9E05E1D71438F936A35751C1A962948260|title=Movie Review - Beverly Hills Cop - FILM: MURPHY IN 'BEVERLY HILLS COP' - NYTimes.com|newspaper=The New York Times|date=2012-10-03|access-date=2017-08-27|url-status=bot: unknown|archiveurl=https://web.archive.org/web/20121003041657/http://movies.nytimes.com/movie/review?res=9E05E1D71438F936A35751C1A962948260|archivedate=2012-10-03|last1=Maslin|first1=Janet}}&lt;/ref&gt; and [[Richard Schickel]].&lt;ref&gt;{{Cite magazine|url=http://content.time.com/time/magazine/article/0,9171,955391,00.html|title=Cinema: Eddie Goes to Lotusland|last=Schickel|first=Richard|date=1984-12-10|magazine=Time|access-date=2017-08-27|language=en-US|issn=0040-781X}}&lt;/ref&gt; In spite of much of the film's comedy having been improvised by the actors,&lt;ref&gt;{{Cite news|url=http://mentalfloss.com/article/60484/15-things-you-may-not-have-known-about-beverly-hills-cop|title=15 Things You May Not Have Known About ''Beverly Hills Cop''|date=2015-10-12|access-date=2017-08-27|language=en}}&lt;/ref&gt; Petrie's contributions earned him accolades in the form of [[Academy Awards|Academy Award]] and [[Edgar Award]] nominations.

706 In 2006, [[Sylvester Stallone]] expressed his desire to direct a film about ''Musa Dagh'', according to Professor Savaş Eğilmez of [[Atatürk University]]. An e-mail campaign sponsored by the Armenian-genocide denying Foundation for the Struggle Against Baseless Allegations of Genocide (ASİMED), pressured Stallone into not proceeding with the film.&lt;ref&gt;&quot;[http://www.todayszaman.com/newsDetail_getNewsById.action?load=detay&amp;link=128034 Gibson urged to reject film with Armenian allegations] {{Webarchive|url=https://web.archive.org/web/20120702033141/http://www.todayszaman.com/newsDetail_getNewsById.action?load=detay&amp;link=128034 |date=2012-07-02 }}.&quot; ''[[Today's Zaman]]'', November 27, 2007.&lt;/ref&gt; In early 2009, reports surfaced that [[Mel Gibson]] was also considering directing a documentary and appearing in the adaptation of Werfel's novel but was dissuaded after receiving 3,000 e-mails from a Turkish pressure group.&lt;ref&gt;"[http://www.asbarez.com/60058/mel-gibson-not-filming-armenian-genocide-documentary/ Mel Gibson Not Filming Armenian Genocide Documentary]." ''[[Asbarez]]''. February 3, 2009.&lt;/ref&gt;

707 | Episode: &quot;Sylvester Stallone/Jerry Springer/Tim Love&quot;

708 *[[Sylvester Stallone]] (December 13, 1990)

709 *''[[First Blood]]'' (1993 Fuji TV edition) - [[John Rambo]] ([[Sylvester Stallone]])&lt;ref&gt;{{cite web |url= https://www.hmv.co.jp/artist_ランボー_000000000101734/item_ランボー-4K レストア版_10419173|title= ランボー4K レストア版|access-date= October 5, 2021|work= HMV}}&lt;/ref&gt;

710 In 1992, Shore starred in ''[[Encino Man]]'', which was a modest hit. The film's success propelled Shore to star in additional films, albeit increasingly less successful: ''[[Son in Law (film)|Son in Law]]'' (1993), ''[[In the Army Now (film)|In the Army Now]]'' (1994), ''[[Jury Duty (film)|Jury Duty]]'' (1995), and ''[[Bio-Dome]]'' (1996). All five films received sharply negative reviews, with the last three each holding a rating below 10% at [[Rotten Tomatoes]]. In addition, each of the films grossed less at the box office than the one before. Describing Shore's performances in these films, film critic [[Roger Ebert]] wrote, &quot;Shore bypasses all categories to achieve a kind of transcendent fingernails-on-the-blackboard effect.&quot;&lt;ref&gt;{{cite web |last1=Ebert | first1=Roger |title=Jury Duty |url=https://www.rogerebert.com/reviews/jury-duty-1995 |website=RogerEbert.com |publisher=Ebert Digital LLC |access-date=2021-02-20}}&lt;/ref&gt; In 1997, Shore starred in the eponymous TV show ''[[Pauly]]'', which was cancelled after five episodes aired. Shore made a [[cameo appearance]] in the American [[rock music|rock]] band [[Limp Bizkit]] music video &quot;[[N 2 Gether Now]]&quot;&lt;ref&gt;{{cite book|last=Devenish|first= Colin |year=2000|url=https://archive.org/details/limpbizkit0000deve |url-access=registration|title=Limp Bizkit|publisher=[[St. Martin's Griffin]]|page= [https://archive.org/details/limpbizkit0000deve/page/148 148]|isbn=978-0-312-26349-2}} {Google books}.&lt;/ref&gt;&lt;ref&gt;{{cite AV media |url=https://www.youtube.com/watch?v=tMLetx7SjCw | archive-url=https://ghostarchive.org/varchive/youtube/20211107/tMLetx7SjCw| archive-date=2021-11-07 | url-status=live|title=N 2 Gether Now| format=Flash video| publisher=[[Limp Bizkit]] [[VEVO]] official YouTube channel}}{{cbignore}}&lt;!-- licensed content--&gt;&lt;/ref&gt; as a pizza deliveryman and a briefer appearance in &quot;[[Break Stuff]]&quot;.&lt;ref&gt;{{cite AV media |url=https://www.youtube.com/watch?v=ZpUYjpKg9KY | archive-url=https://ghostarchive.org/varchive/youtube/20211107/ZpUYjpKg9KY| archive-date=2021-11-07 | url-status=live|title=Break Stuff | format=Flash video |publisher= [[Limp Bizkit]] [[VEVO]] official YouTube channel}}{{cbignore}}&lt;!-- licensed content--&gt;&lt;/ref&gt; The [[Golden Raspberry Awards]] has recognized Shore's film performances several times, awarding him Worst New Star of the Year for ''Encino Man'', Worst Actor of the Year for ''Bio-Dome'', and Worst New Star of the Decade for the 1990s, and nominating him for Worst Actor of the Century (which he lost to [[Sylvester Stallone]]).&lt;ref&gt;{{cite web |last1=Dembrow |first1=Dylan |title=15 Actors Who Have Won The Most Razzie Awards |url=https://screenrant.com/actors-who-won-most-razzie-awards-ever/ |website=Screen Rant |date=26 February 2017 |publisher=screenrant.com |access-date=2021-02-10}}&lt;/ref&gt;

711 After ''T.H.E. Cat'', Loggia, an actor with a long history of film and television credits, went on to star in a number of high-profile Hollywood films, including ''[[An Officer and a Gentleman]]'', ''[[Scarface (1983 film)|Scarface]]'', Sylvester Stallone's ''[[Over the Top (1987 film)|Over the Top]]'', the [[Tom Hanks]] comedy ''[[Big (film)|Big]]'', and the science fiction action film ''[[Independence Day (1996 film)|Independence Day]]''.&lt;ref name=&quot;lat&quot;&gt;{{cite web|url=http://www.calendarlive.com/movies/reviews/cl-movie960716-1,2,1312906.story|title=Independence Day review|author=[[Kenneth Turan]]|date=July 2, 1996|work=[[Los Angeles Times]]|archive-url=https://web.archive.org/web/20080619051208/http://www.calendarlive.com/movies/reviews/cl-movie960716-1%2C2%2C1312906.story|archive-date=June 19, 2008|url-status=dead|access-date=July 8, 2008|df=mdy-all}}&lt;/ref&gt; In 1985, Loggia was nominated for an Academy Award for Best Supporting Actor for his portrayal of crusty private detective Sam Ransom in the thriller ''[[Jagged Edge (film)|Jagged Edge]],'' and had the starring role in another NBC series, ''[[Mancuso, FBI]]'', for which he was nominated for an Emmy in 1989.&lt;ref&gt;{{Cite web|url=https://www.emmys.com/shows/mancuso-fbi|title=Mancuso Fbi}}&lt;/ref&gt;

712 * [[Sylvester Stallone]] as [[Kublai Khan]] (Billy West)

713 Another famous guest of the Julien was [[Sylvester Stallone]]. Stallone stayed at the Julien Inn during the filming of the movie ''[[F.I.S.T. (film)|F.I.S.T.]]''.

714 The film was not well received by the critics, with ''[[The New York Times]]'' calling it &quot;not a good film&quot;.&lt;ref&gt;{{cite web|url=https://movies.nytimes.com/movie/review?res=9407E7D91438F93AA15750C0A963948260|title=Movie Review - SCREEN: 'KING DAVID,' A BIBLICAL EPIC|last=Canby|first=Vincent|newspaper=[[The New York Times]]|date=March 29, 1985}}&lt;/ref&gt; Review aggregate [[Rotten Tomatoes]] gave the film a 'rotten' 8% rating.&lt;ref&gt;{{cite web|url=http://www.rottentomatoes.com/m/king_david/|title=King David|publisher=[[Rotten Tomatoes]]}}&lt;/ref&gt; Richard Gere's performance in the film earned him a [[Golden Raspberry Award]] nomination for Worst Actor, which he lost to [[Sylvester Stallone]] for ''[[Rambo: First Blood Part II]]'' and ''[[Rocky IV]]''.

715 * [[Sylvester Stallone]]

716 [[Sylvester Stallone]] attended the fight, as well as former world Heavyweight champion [[Mike Tyson]] and Jr. Welterweight contender and interim WBA Jr. Welterweight champion [[Vivian Harris]].

717 * [[Sylvester Stallone]]

718 * Sylvester Stallone

719 '''''Rambo III''''' is a 1988 American [[action film]] directed by [[Peter MacDonald (director)|Peter MacDonald]] and co-written by [[Sylvester Stallone]], who also reprises his role as [[Vietnam War]] veteran [[John Rambo]]. A sequel to ''[[Rambo: First Blood Part II]]'' (1985), it is the third installment in the [[Rambo (franchise)|''Rambo'' franchise]].

720 * [[Sylvester Stallone]] as [[John Rambo|John J. Rambo]]

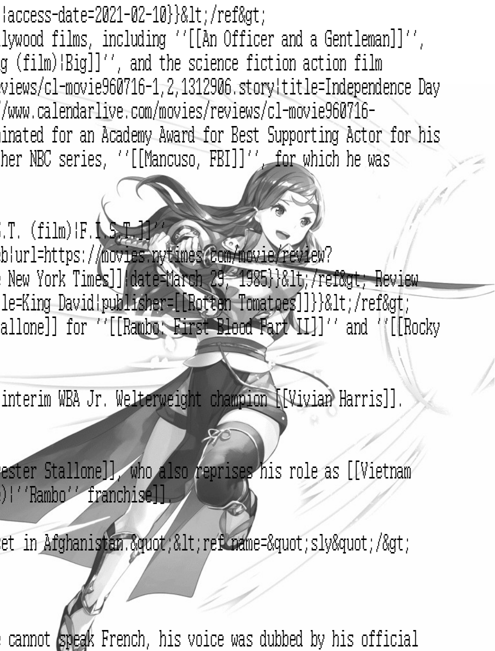
721 Sylvester Stallone later said his original premise of the film &quot;was more in keeping with the theme of ''[[Tears of the Sun]]'', but set in Afghanistan.&quot;&lt;ref name=&quot;sly&quot;/&gt;

722 |rowspan=2|[[Sylvester Stallone]]

723 {{Sylvester Stallone}}

724 [[Category:Films with screenplays by Sylvester Stallone]]

725 * [[Sylvester Stallone]] had an uncredited [[Cameo appearance|cameo]] at the beginning of the film as a passenger to the airport. Since he cannot speak French, his voice was dubbed by his official



French dub-over artist, [[Alain Dorval]].

726 ¦ relatives = [[Sylvester Stallone]] (brother)&lt;br/&gt;[[Sage Stallone]] (nephew)&lt;br /&gt;[[Sistine Stallone]] (niece)

727 '''Frank Stallone Jr.''' (born July 30, 1950)&lt;ref name=&quot;allmusic&quot;&gt;{{cite web¦url=https://www.allmusic.com/artist/frank-stallone-mn0000792942/biography¦title=Frank Stallone Biography¦last=Carpenter¦first=Bil¦publisher=allmusic¦accessdate=November 6, 2022}}&lt;/ref&gt; is an American actor and musician. He is the younger brother of actor [[Sylvester Stallone]]{{Citation needed¦date=August 2024}} and has written music for Sylvester's movies.{{Citation needed¦date=August 2024}} His song &quot;[[Far from Over (Frank Stallone song)¦Far from Over]]&quot; appeared in the 1983 film ''[[Staying Alive (1983 film)¦Staying Alive]]''{{Citation needed¦date=August 2024}} and was included in [[Staying Alive (1983 film)#Soundtrack¦the film's soundtrack album]].{{Citation needed¦date=August 2024}} The song reached number ten on the US [[Billboard Hot 100¦''Billboard'' Hot 100]]{{Citation needed¦date=August 2024}} and received a Golden Globe nomination for Stallone{{Citation needed¦date=August 2024}}, while the album itself, consisting of Stallone and various other artists, received a Grammy nomination.{{Citation needed¦date=August 2024}}

728 Stallone was born in [[Maryland]] and grew up in [[Philadelphia]].&lt;ref&gt;{{Cite web ¦title=Biography ¦url=https://www.frankstallone.com/biography ¦access-date=2021-08-21 ¦website=Frank Stallone (personal website) ¦language=en}}&lt;/ref&gt; He is the younger son of [[Jackie Stallone¦Jacqueline Stallone]]{{Citation needed¦date=August 2024}} (née Labofish; 1921-2020&lt;ref&gt;{{Cite web ¦last=Shafer ¦first=Ellise ¦date=2020-09-22 ¦title=Jackie Stallone, Sylvester Stallone's Mother and Celebrity Astrologist, Dies at 98 ¦url=https://variety.com/2020/film/obituaries-people-news/jackie-stallone-dead-sylvester-stallone-mom-1234778583/ ¦access-date=2024-08-03 ¦website=Variety ¦language=en-US}}&lt;/ref&gt;), an [[astrologer]]{{Citation needed¦date=August 2024}}, former [[Dance¦dancer]]{{Citation needed¦date=August 2024}}, and promoter of women's [[professional wrestling]]{{Citation needed¦date=August 2024}}, and [[Frank Stallone Sr.]] (1919-2011),{{Citation needed¦date=August 2024}} a hairdresser.{{Citation needed¦date=August 2024}} His father was an Italian immigrant{{Citation needed¦date=August 2024}}, and his mother's family was [[France¦French]] from [[Brittany]]{{Citation needed¦date=August 2024}} and also of Eastern European descent.{{Citation needed¦date=August 2024}} Stallone's brother is actor [[Sylvester Stallone]].{{Citation needed¦date=August 2024}} In his teen years, he went to [[Abraham Lincoln High School (Philadelphia)¦Lincoln High School]] in Northeast [[Philadelphia]].&lt;ref name=Stallone&gt;{{cite tweet ¦title=Just found my high school ring. Lincoln High School Philadelphia. https://www.instagram.com/p/BD5IxymHOlzHU_q30k8ZOO2C9VqwpT4gbanCzM0/ ¦user=Stallone ¦author=Frank Stallone ¦number=717996306312990720 ¦date=April 7, 2016}}&lt;/ref&gt;

729 A documentary directed and produced by [[Derek Wayne Johnson]] entitled ''STALLONE: Frank, That Is'' about the life, career, and survival of Frank Stallone was released in 2021. The documentary features interviews with [[Sylvester Stallone]], [[Arnold Schwarzenegger]], [[Billy Zane]], [[Geraldo Rivera]], [[Joe Mantegna]] and many others.&lt;ref&gt;Gonzalez, Bobby. [https://ktla.com/2018/02/19/spoken-dreams-derek-wayne-johnson-filmmaker/ &quot;Spoken Dreams: Derek Wayne Johnson, Filmmaker&quot;], ''KTLA'', Los Angeles, CA, February 19, 2018. Retrieved on August 21, 2018.&lt;/ref&gt;

730 [[Category:Family of Sylvester Stallone]]</text>

731 In 1982, Vajna was a founder and then president of the American Film Marketing Association. During that same year, Vajna and Kassar made their film production debut with ''[[First Blood¦Rambo: First Blood]]'', starring [[Sylvester Stallone]]. ''First Blood'' was a success, grossing $120 million internationally. ''[[Rambo: First Blood Part II]]'' was released in 1985, generating more than $300 million worldwide.

732 A Russian pool configuration can be seen in “Tulsa King” starring Sylvester Stallone in Season 1, episode 5, while making a phone call, he walks around a pool table with all-white balls racked, and a white cue ball.

733 In the 1993 film ''[[Demolition Man (film)¦Demolition Man]]'', Lenina Huxley ([[Sandra Bullock]]) mentions a &quot;Schwarzenegger [[Presidential library system¦Library]]&quot; to John Spartan ([[Sylvester Stallone]]) and attempts to explain a &quot;61st Amendment&quot; to him. This implied that, at some point before 2032, legislation similar to the &quot;Equal Opportunity to Govern Amendment&quot; was passed in the film's timeline, allowing Schwarzenegger to run for president of the United States.

734 The script was based on an original screenplay by future U.S. senator [[Jim Webb¦James Webb]]. It had previously been in development at [[Universal Pictures]] for about ten years&lt;ref&gt;{{Cite web¦url=https://www.usni.org/magazines/proceedings/2000/june/they-all-hated-rules-engagement¦title=They All Hated Rules of Engagement'¦date=June 2000¦access-date=2022-11-07¦archive-date=2022-11-08¦archive-url=https://web.archive.org/web/20221108004455/https://www.usni.org/magazines/proceedings/2000/june/they-all-hated-rules-engagement¦url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://www.bbc.co.uk/films/2000/08/09/william_friedkin1_interview.shtml¦title=Films - interview - William Friedkin¦access-date=2023-06-09¦archive-date=2023-04-29¦archive-url=https://web.archive.org/web/20230429153625/https://www.bbc.co.uk/films/2000/08/09/william_friedkin1_interview.shtml¦url-status=live}}&lt;/ref&gt; before being acquired by [[Paramount Pictures]], where the script was further developed under producer [[Scott Rudin]], with [[Sylvester Stallone]] in talks to star in the film.&lt;ref&gt;{{Cite web¦url=https://variety.com/1997/voices/columns/two-pix-for-twohy-nightfall-havoc-1116678689/¦title=Two pix for Twohy: 'Nightfall' &amp; 'Havoc'¦date=1997-08-14¦magazine=[[Variety (magazine)¦Variety]]¦access-date=2022-11-07¦archive-date=2022-11-07¦archive-url=https://web.archive.org/web/20221107094808/https://variety.com/1997/voices/columns/two-pix-for-twohy-nightfall-havoc-1116678689/¦url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://variety.com/1997/voices/columns/wb-locks-rock-for-role-in-lethal-weapon-4-111662482/¦title=WB locks Rock for role in 'Lethal Weapon 4'¦date=1997-10-07¦magazine=[[Variety (magazine)¦Variety]]¦access-date=2022-11-07¦archive-date=2022-11-07¦archive-url=https://web.archive.org/web/20221107094823/https://variety.com/1997/voices/columns/wb-locks-rock-for-role-in-lethal-weapon-4-111662482/¦url-status=live}}&lt;/ref&gt; [[William Friedkin]] was hired to direct,&lt;ref&gt;{{Cite web¦url=https://variety.com/1998/film/news/general-to-par-for-2-mil-1117467518/amp/¦title=General' to Par for $2 mil¦date=1998-02-08¦magazine=[[Variety (magazine)¦Variety]]¦access-date=2022-11-07}}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://variety.com/1998/film/news/friedkin-set-to-tell-truth-1117468626/¦title=Friedkin set to tell 'Truth'¦date=1998-03-10¦magazine=[[Variety (magazine)¦Variety]]¦access-date=2022-11-07¦archive-date=2022-11-07¦archive-url=https://web.archive.org/web/20221107094823/https://variety.com/1998/film/news/friedkin-set-to-tell-truth-1117468626/¦url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://bombreport.com/yearly-breakdowns/2000-2/rules-of-engagement/¦title=Rules Of Engagement¦date=2000¦website=www.bombreport.com¦access-date=2024-02-11¦archive-date=2024-02-29¦archive-url=https://web.archive.org/web/20240229060009/https://bombreport.com/yearly-breakdowns/2000-2/rules-of-engagement/¦url-status=live}}&lt;/ref&gt; but had trouble collaborating with Webb on script rewrites. Rudin passed the project over to [[Richard Zanuck]], who then hired [[Stephen Gaghan]] to rewrite the screenplay, Gaghan dived into the project, reading the [[Tim O'Brien (author)¦Tim O'Brien]]' novels &quot;[[The Things They Carried]]&quot; and &quot;[[Going After Cacciato]]&quot;, and watching the 1957 film &quot;[[Paths of Glory]]&quot;.&lt;ref&gt;{{Cite book¦url=https://books.google.com/books?id=J75iAgAAQBAJ&amp;dq=stephen+gaghan+rules+of+engagement&amp;pg=PA85¦title=FilmCraft: Screenwriting¦last=Grierson¦first=Tim¦date=2013¦publisher=Taylor &amp; Francis¦isbn=9781136070624¦language=en¦access-date=2023-06-09}}&lt;/ref&gt; Webb hated Gaghan's work and frustrated the filmmaker's attempts to receive cooperation from the Department of Defense,{{citation needed¦date=April 2019}} which was eventually obtained nonetheless.

735 * Bob &quot;Bull&quot; Hurley, an arm-wrestler from Sylvester Stallone's film ''[[Over the Top (1987 film)¦Over the Top]]''

736 ¦ Backed by [[Sylvester Stallone]], [[Bruce Willis]], [[Demi Moore]], and [[Arnold Schwarzenegger]]

737 Le began [[Wrestling]] competitively at age 14. After being inspired to box by [[Sylvester Stallone]]´s ''[[Rocky]]'', Le graduated and earned [[All-American]] honors in Wrestling his junior year at San Jose High School.&lt;ref&gt;{{Cite web ¦last=Robertson , Special to the Chronicle ¦first=Jordan ¦date=February 1, 2012 ¦orig-date=February 1, 2012 ¦title=Over the Top / At whatever he's taken on, Cung Le has been successful - from kickboxing to fatherhood ¦url=https://www.sfgate.com/bayarea/article/Over-the-Top-At-whatever-he-s-taken-on-Cung-Le-2929714.php ¦url-status=live ¦archive-url=https://web.archive.org/web/20210211085731/https://www.sfgate.com/bayarea/article/Over-the-Top-At-whatever-he-s-taken-on-Cung-Le-2929714.php ¦archive-date=February 11, 2021 ¦access-date=September 24, 2024 ¦website=The San Francisco Gate ¦language=English}}&lt;/ref&gt; He went on to wrestle for [[West Valley College]] in [[Saratoga, California]], and won the California Junior College State Championship in the 158&amp;nbsp;lb weight class in 1990 also earning junior college All-American honors. Le also is practiced in a variety of martial arts such as [[Vovinam]], [[Judo]], [[Karate]], [[Muay Thai]], [[Boxing]], [[Kuntao]], [[Sambo (martial art)¦Sambo]], [[Brazilian Jiu-Jitsu]], and [[Sanda (sport)¦Sanda]] - which became his main focus. {{citation needed¦date=August 2015}}. Le held a professional Kickboxing record of 17-0 and is a three-time world champion in Kickboxing.

738 In the [[Syfy¦Sci-Fi Channel]] original film ''[[Locusts: The 8th Plague]]'', she played the lead female role Vicky. She also appeared in the straight-to-DVD film ''[[8mm 2]]'' as Lynn. The film was originally called ''The Velvet Side of Hell'' and was not supposed to be a sequel to ''[[8MM (film)¦8MM]]''. Benz appeared in episodes of ''[[Supernatural (American TV series)¦Supernatural]]'', ''[[CSI: Miami]]'', ''[[Law &amp; Order]]'', and ''[[CSI: Crime Scene Investigation]]''. She had a supporting role in the [[Sweden¦Swedish]] independent film ''[[Kill Your Darlings (2006 film)¦Kill Your Darlings]]'' (2006), a lead role in the television film ''Circle of Friends'' (2006), and joined the cast of the TV show ''[[Dexter (TV series)¦Dexter]]'' as [[List of Dexter characters#Rita Bennett¦Rita Bennett]], [[Dexter Morgan]]'s girlfriend and eventual wife.&lt;ref&gt;{{cite web¦url=https://www.bloody-disgusting.com/news/17745 ¦title=SCREAM '09: Julie Benz, Jennifer Carpenter on 'Dexter', More! ¦website=[[Bloody-disgusting.com]]¦date=October 18, 2009 ¦access-date=October 20, 2010}}&lt;/ref&gt; Benz played a lead role in the fifth film of the popular ''Saw'' horror franchise, ''[[Saw V]]'', as [[List of Saw characters#Brit¦Brit]], a real estate developer who is one of [[Jigsaw (Saw character)¦Jigsaw]]'s five victims. Benz had a supporting role in ''[[Punisher: War Zone]]'' as Angela and co-starred with [[Sylvester Stallone]] in ''[[Rambo (2008 film)¦Rambo]]'' (2008), the fourth film of the [[Rambo (film series)¦series]].

739 In 2010, West Coast Customs designed three identical customized 1955 [[Ford F-100]] [[pick up truck]]s that were prominently featured in the film ''[[The Expendables (2010 film)¦The Expendables]]''.&lt;ref&gt;{{Cite news¦url=http://link.galegroup.com/apps/doc/A245542108/GPS?sid=wikipedia¦title=The expendables' F-100: LMC truck teams up with West Coast customs for a wild ride¦last=Mata¦first=John Jr.¦date=August 23, 2010¦work=Truckin'¦publisher=The Enthusiast Network¦url-access=subscription ¦access-date=January 22, 2017¦via=Gale (General OneFile)}}&lt;/ref&gt; One of these cars was for [[Sylvester Stallone]], who sold it at auction in 2011 for $132,000.&lt;ref&gt;{{Cite news¦url=http://www.autoevolution.com/news/sylvester-stallone-s-expendables-truck-sells-for-132000-30195.html¦title=Sylvester Stallone's &quot;Expendables&quot; Truck Sells for $132,000¦last=Popa¦first=Bogdan¦date=January 25, 2011¦newspaper=autoevolution¦language=en-us¦access-date=January 22, 2017}}&lt;/ref&gt; In 2015, WCC created a car by combining the body of a [[Ford Maverick]] and the internals of a [[Ford Ranger (North America)¦Ford Ranger]] for another film, ''[[Mad Max: Fury Road]]''.&lt;ref&gt;{{Cite news¦url=http://www.autoevolution.com/news/west-coast-customs-made-a-magnum-opus-car-for-mad-max-game-and-its-apocalyptic-video-99460.html¦title=West Coast Customs Made a Magnum Opus Car for Mad Max Game, and It's Apocalyptic¦last=Ungureanu¦first=Ionut¦date=September 1, 2015¦newspaper=autoevolution¦language=en-us¦access-date=January 22, 2017}}&lt;/ref&gt;

740 In 1981, Wark was one of several Ipswich players who appeared alongside stars including [[Bobby Moore]] and [[Pelé]] in the [[Second World War]] football film ''[[Escape to Victory]]'', which starred [[Sylvester Stallone]], [[Michael Caine]] and [[Max von Sydow]].&lt;ref name=&quot;escape&quot;&gt;{{cite news¦ url = http://www.timesonline.co.uk/tol/sport/football/scotland/article6005964.ece¦ title = Terry Butcher calls for Scott Brown to walk in John Wark's footsteps ¦ date = 31 March 2009¦ access-date = 30 April 2009 ¦work=The Times ¦ first = Phil ¦ last = Gordon ¦ location=London}}{{dead link¦date=September 2024¦bot=medic}}{{cbignore¦bot=medic}}&lt;/ref&gt; Wark played a character called Arthur Hayes,&lt;ref&gt;Wark, p. 157&lt;/ref&gt; but his only line of dialogue was dubbed due to his broad [[Glaswegian dialect¦Glaswegian accent]].&lt;ref name=escape/&gt;

741 :::These claims have been swallowed by the rich and the famous. Among Aslan's many clients were John F. Kennedy, Marlene Dietrich, Kirk Douglas, and Salvador Dali. Sylvester Stallone also reportedly takes GH3 shots to remain sexy.&quot; [http://www.netasia.net/users/truehealth/Gerovital.htm]

742 ¦ [[Sylvester Stallone]] ¦¦ Motion pictures ¦¦ 6712 Hollywood Boulevard ¦¦ {{birth date¦1984¦06¦14}}

743 &quot;'''GLC: The Carnage Continues'''&quot; is an episode of the British [[television]] [[comedy]] series ''[[The Comic Strip¦The Comic Strip Presents...]]'' broadcast on [[BBC2]] in 1990. It parodied a Hollywood telling of the 1980s takeover of the [[Greater London Council]] by [[Ken Livingstone]] and the subsequent disbanding of that body by Prime Minister [[Margaret Thatcher]], re-imagining the story as a [[Charles Bronson]] / [[Sylvester Stallone]]-style action movie. It is a [[spiritual successor]] to ''[[The Strike]]'', which involved the creation of a Hollywood version of

the [[UK miners' strike (1984-1985)¦1984 miners' strike]].&lt;ref name=bush&gt;{{cite web ¦title=Kate Bush Once Wrote A Song About Ken Livingstone ¦url=http://www.lbc.co.uk/kate-bush-once-wrote-a-song-about-ken-livingstone-88090 ¦date=28 March 2014 ¦accessdate=1 May 2016 ¦url-status=dead ¦archiveurl=https://web.archive.org/web/20150919090621/http://www.lbc.co.uk/kate-bush-once-wrote-a-song-about-ken-livingstone-88090 ¦archivedate=19 September 2015 ¦df=dmy-all }}&lt;/ref&gt;

744 [[Eddie Barth]] and [[Nicholas Saunders (actor)¦Nicholas Saunders]] make their theatrical film debuts as the characters Paul and Douglas, while comedian [[Conrad Bain]] plays Semple and actor [[Allen Garfield]] plays the Man on Cross. Uncredited appearances include [[Sylvester Stallone]] as a subway thug #1, [[Mary Jo Catlett]] as a woman in a hotel lobby&lt;ref name=&quot;imdbcast&quot;&gt;{{cite web ¦title=Full cast and crew for Bananas ¦url=https://www.imdb.com/title/tt0066808/fullcredits ¦url-status=live ¦archive-url=https://web.archive.org/web/20111113132022/http://www.imdb.com/title/tt0066808/fullcredits ¦archive-date=November 13, 2011 ¦access-date=May 11, 2012 ¦website=Internet Movie Database}}&lt;/ref&gt; and [[Tino García]] in an undisclosed role.

745 DiCola was born in 1957 in [[Lancaster, Pennsylvania]], to Italian-American parents. After majoring in percussion in college, DiCola began his professional music career upon moving to California in 1981. One of his early credits as a session musician was his synthesizer performance on [[Juice Newton]]'s albums ''Dirty Looks'' and ''Old Flame'' released in 1983 and 1985, respectively. His first major break came when he was chosen to co-write several songs on the ''[[Staying Alive (1983 film)¦Staying Alive]]'' soundtrack (1983) with [[Frank Stallone]].&lt;ref&gt;{{Cite web ¦title=Vince DiCola Songs, Albums, Reviews, Bio &amp; More ¦url=https://www.allmusic.com/artist/vince-dicola-mn0000211848 ¦access-date=2023-09-19 ¦website=AllMusic ¦language=en}}&lt;/ref&gt; Following his work on the ''Staying Alive'' project, he was recruited by [[Sylvester Stallone]] to write the original score for ''[[Rocky IV]]'' (1985), followed by scoring ''[[The Transformers: The Movie]]'' (1986). Since then, he has been active in the capacity of producer, recorded and performed as both a solo artist and member of several bands, and continues to work as a session musician and contributor to film soundtracks as composer and performer.

746 In 2019, Vega appeared in [[Televisa]] to portray the lead character Catalina Creel in the miniseries ''[[Cuna de lobos (2019 TV series)¦Cuna de Lobos]]'', based on the [[Cuna de lobos¦1986 original telenovela of the same title]]. The same year, she portrayed an independent Mexican reporter in ''Rambo: Last Blood'', in which she co-starred with [[Sylvester Stallone]].

747 ''Pa-Pa-Pa-Pa-Puffy'' featured guests such as [[Lenny Kravitz]], [[Sylvester Stallone]], [[Harrison Ford]], [[Aerosmith]], and rock band [[Garbage (band)¦Garbage]].&lt;ref&gt;{{Cite web¦url=https://www.cherryred.co.uk/product/hi-hi-puffy-amiyumi-expanded-european-edition/¦title=Hi Hi Puffy Amiyumi (Expanded European Edition)¦website=Cherry Red Records¦language=en-US¦access-date=2019-02-07}}&lt;/ref&gt;

748 During the 1970s, Assante was a regular on two NBC soap operas, ''[[How to Survive a Marriage]]'' as Johnny McGhee and ''[[The Doctors (1963 TV series)¦The Doctors]]'' as Mike Powers. His first film was ''[[The Lords of Flatbush]]'' (1974), although his work did not appear in the final cut of the film and the end credits misspelled his last name as Assanti. He starred in [[Prophecy (film)¦''&lt;u&gt;Prophecy&lt;/u&gt;'']] (1979). His first on-screen role was playing Sylvester Stallone's brother in ''[[Paradise Alley]]'', and a role that brought him greater attention came in 1980's ''[[Private Benjamin (1980 film)¦Private Benjamin]]'' as a handsome Frenchman who becomes the love interest of a U.S. soldier played by [[Goldie Hawn]]. In 1984 he portrayed the playboy violin virtuoso Maximillian Stein in the [[Dudley Moore]] comedy vehicle ''[[Unfaithfully Yours (1984 film)¦Unfaithfully Yours]]''.

749 Other film appearances include one as a Cuban bandleader in ''[[The Mambo Kings]]'' opposite [[Antonio Banderas]] and in the adaptation of the science-fiction story ''[[Judge Dredd (film)¦Judge Dredd]]'' with [[Sylvester Stallone]], his co-star in ''[[Paradise Alley]]''. He appeared in the 2007 film ''[[American Gangster (film)¦American Gangster]]'' with [[Denzel Washington]] and [[Russell Crowe]]. He had a recurring guest star role in several episodes of ''[[NCIS (TV series)¦NCIS]]'', playing the international arms dealer [[List of NCIS characters#René Benoit¦René Benoit]]. He has also appeared in the 2009 horror film ''[[Smile (2009 film)¦Smile]]'' as the mysterious Tollinger and portrayed Ernie in ''[[Sugar (2022 film)¦Sugar]]'' (marketed on [[Lifetime (TV network)¦Lifetime]] as ''Danger Below Deck''). Outside of the U.S., he has participated in several film projects and humanitarian endeavors in [[Romania]], [[Bulgaria]], [[North Macedonia]], [[Serbia]], [[Croatia]], [[Bosnia-Herzegovina]], [[Kazakhstan]], and [[Uzbekistan]].

750 Coming out of college, Bavaro was known mostly for his blocking ability and was expected to play the role of a run blocking tight end as a professional.&lt;ref&gt;[[Frank Litsky¦Litsky, Frank]]. [https://select.nytimes.com/search/restricted/article?res=F20F1FF9385F0C718EDDAC0894DD484D81 New Giant in Familiar Role] {{Webarchive¦url=https://web.archive.org/web/20211107093113/https://www.nytimes.com/1985/05/22/sports/new-giant-in-familiar-role.html ¦date=November 7, 2021 }}, ''The New York Times'', May 22, 1985, accessed May 14, 2007.&lt;/ref&gt; He earned the nickname &quot;[[John Rambo¦Rambo]]&quot; early in his rookie season due to his intense playing style, quiet personality, and physical resemblance to [[Sylvester Stallone]].&lt;ref name=&quot;ND&quot;/&gt; After starting tight end [[Zeke Mowatt]] suffered a season-ending injury before the start of the 1985 season, coach [[Bill Parcells]], who had called Bavaro the most impressive rookie during training camp, installed Bavaro as the starter.&lt;ref name=&quot;jeans&quot;&gt;Litsky, Frank. [https://select.nytimes.com/search/restricted/article?res=F50811FE3E5F0C728CDDA00894DD484D81 Mowatt Sidelined for Entire Season] {{Webarchive¦url=https://web.archive.org/web/20211107093131/https://www.nytimes.com/1985/09/01/sports/mowatt-sidelined-for-entire-season.html ¦date=November 7, 2021 }}, ''The New York Times'', September 1, 1985, accessed May 14, 2007.&lt;/ref&gt; Bavaro finished his rookie season with 37 receptions, 511&amp;nbsp;yards, and 4 [[touchdown]]s.&lt;ref name=&quot;dbf&quot;&gt;[http://www.databasefootball.com/players/playerpage.htm?ilkid=BAVARMAR01 Mark Bavaro] {{webarchive¦url=https://web.archive.org/web/20060328102824/http://www.databasefootball.com/players/playerpage.htm?ilkid=BAVARMAR01 ¦date=March 28, 2006 }}, databasefootball.com, accessed May 14, 2007.&lt;/ref&gt; He also set a team record with 12 receptions in one game during [[quarterback]] [[Phil Simms]]' 513&amp;nbsp;yard passing effort against the [[Cincinnati Bengals]] on October 13, 1985.&lt;ref name=&quot;record&quot;&gt;[https://select.nytimes.com/search/restricted/article?res=FA081EF93A5D0C778DDDA90994DD484D81 A Great Day Wasn't Enough] {{Webarchive¦url=https://web.archive.org/web/20211107093113/https://www.nytimes.com/1985/10/14/sports/a-great-day-wasn-t-enough.html ¦date=November 7, 2021 }}, ''The New York Times'', October 14, 1985, accessed May 14, 2007.&lt;br /&gt;* [http://www.databasefootball.com/boxscores/scheduleyear.htm?yr=1985&amp;lg=nfl 1985 NFL Season Scores, Schedules and Playoffs] {{webarchive¦url=https://web.archive.org/web/20070408010955/http://www.databasefootball.com/boxscores/scheduleyear.htm?lg=nfl&amp;yr=1985 ¦date=April 8, 2007 }}, databasefootball.com, accessed May 14, 2007.&lt;/ref&gt; After the game, which the Giants lost 35-30, Bavaro responded in the low-key manner that would typify his career, &quot;[i]t was nothing special, the plays were the same stuff. I don't know what they did. I just caught a lot of balls. I'd rather win, that's all.&quot;&lt;ref name=&quot;record&quot;/&gt; Bavaro was named to the [[PFWA All-Rookie Team]] for his performance during the season.&lt;ref name=&quot;blocker&quot;/&gt;

751 Neil Cuthbert's script had been in development with producer [[Martin Bregman]] since at least 1983 under the shorter title ''Pluto Nash''. Directors [[Rick Rosenthal]] and [[Peter Faiman]] were attached to the project at different times. Various stars including [[Harrison Ford]] were rumored to have considered doing the film. [[Eddie Murphy]] became attached to the film while it was in development at [[Castle Rock Entertainment]]. The company, which had previously worked with director [[Ron Underwood]] on ''[[City Slickers]]'', sent Underwood the script. Underwood signed on to the film after the release of his film ''[[Mighty Joe Young (1998 film)¦Mighty Joe Young]]''. Different writers were hired to revise the script over a period of several months, but [[Eddie Murphy]] &quot;kept rejecting the scripts,&quot; according to Underwood. Murphy asked for a &quot;straight&quot; script, like one that would be written for [[Sylvester Stallone]] or Harrison Ford, and said that he would &quot;bring the comedy.&quot; Underwood delivered this to Murphy but believed that the script still had &quot;problems&quot; and hired another writer to continue to work on it during production.&lt;ref name=&quot;slashfilm&quot;&gt;{{cite web¦url=https://www.slashfilm.com/ron-underwood-interview/¦last=Harris¦first=Blake¦title=How Did This Get Made: A Conversation With Ron Underwood, Director of 'Tremors,' 'City Slickers,' and 'The Adventures of Pluto Nash'¦work=[[Slashfilm]]¦date=July 24, 2020}}&lt;/ref&gt;

752 Originally the film was intended to star [[Wesley Snipes]] and [[Jet Li]], and then after them [[Sylvester Stallone]] and [[Vin Diesel]] were going to star. Kaosaynanda's preferred choices were [[Chow Yun-fat]] and [[Jean Reno]], but producers didn't consider them financially viable.&lt;ref name=&quot;:0&quot;&gt;{{Cite web ¦last=III ¦first=Lee B. Golden ¦date=2014-06-02 ¦title=BALLISTIC CINEMATHEQUE: A Word With Wych Kaosayananda - Film Combat Syndicate ¦url=https://filmcombatsyndicate.com/ballistic-cinematheque-word-with-wyc/ ¦access-date=2024-07-21 ¦website=filmcombatsyndicate.com ¦language=en-US}}&lt;/ref&gt;

753 He wrote the 1994 [[horror film]] ''[[In the Mouth of Madness]]'', directed by [[John Carpenter]]. De Luca co-wrote the story for the 1995 [[Judge Dredd (film)¦film adaptation]] of the popular British comic book character [[Judge Dredd]], starring [[Sylvester Stallone]] as the eponymous lead.&lt;ref&gt;{{cite web¦url=https://www.imdb.com/title/tt0113492/fullcredits?ref_=tt_ov_wr#writers¦title=Judge Dredd¦date=June 30, 1995¦access-date=October 31, 2016¦via=IMDb}}&lt;/ref&gt; In 1996, De Luca, was approached by the producers [[Brannon Braga]] and [[Ronald D. Moore]] for ''[[Star Trek: Voyager]]'' and pitched an idea which turned into an opportunity to write an episode for the series, [[Threshold (Star Trek: Voyager)¦''Threshold'']].

754 In 2015 a series of ads were recorded with the action actor [[Sylvester Stallone]] in his role of [[Rocky Balboa]], in this campaign Rocky appears saying the phrase &quot;''Te hace falta ver más box''&quot; (In English ''You need to watch more [[boxing]]'') in which he gives a cold stare to some men that are not acting so &quot;manly&quot; (e.g. listening to pop music or putting on too many clothes for cold weather). In the end Stallone says that &quot;''Somos box''&quot; (&quot;''We are box''&quot;) while retired boxers are shown and &quot;[[Eye of the Tiger¦Eye of the tiger]]&quot; from [[Survivor (band)¦Survivor]]) is played. In Mexico the word is changed to ''Bax'', as Stallone uses the [[American English]] pronunciation for the [[Boxing¦sport]]. At the end of the commercial, Rocky looks moved by the progress of the people he helped only to receive a cold stare from Stallone himself.

755 He was a guest star on a Greek program ''Koita ti Ekanes'' in late-2003 and featured clips from ''Star Academy''. He has published a book called ''I was born Greek: The mythology or The school of life'' ({{lang-el¦Γεννήθηκα Έλληνας: Η Μυθολογία Ή Το Σχολείο Της Ζωής}}). He was the presenter of a show on [[Alpha TV]] called ''Gros Plan'' where he met international stars like [[Celine Dion]], [[Jean Paul Gaultier]], [[Sylvester Stallone]] and [[Helena Paparizou]].&lt;ref&gt;{{ cite web ¦ title = Γκρο Πλαν¦ url = http://www.alphatv.gr/index.asp?a_id=103&amp;tv_id=1414 }} {{in lang¦el}} ''Alpha TV''. Retrieved on May 5, 2008.&lt;/ref&gt;

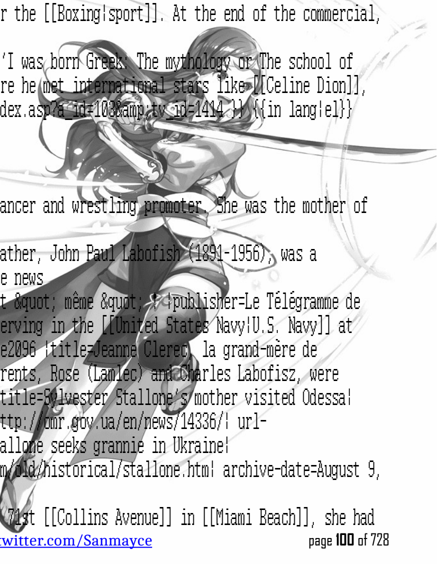
756 ¦ [[Sylvester Stallone]]

757 ¦children   = 3, including [[Sylvester Stallone¦Sylvester]] and [[Frank Stallone¦Frank]]

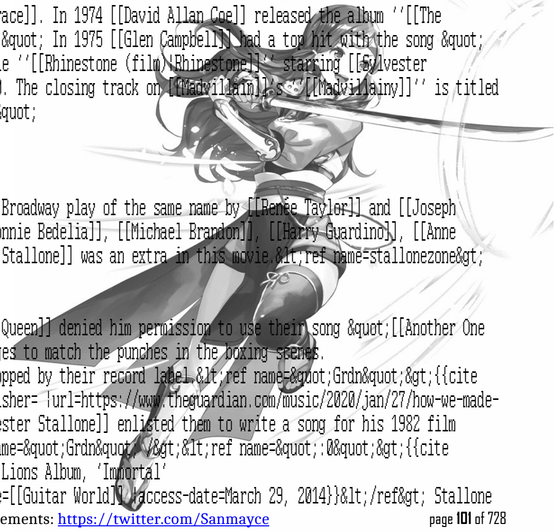
758 '''Jacqueline Frances Stallone''' (''{{nee}}'' '''Labofish'''; November 29, 1921 - September 21, 2020) was an American [[Astrology¦astrologer]], dancer and wrestling promoter. She was the mother of actor [[Sylvester Stallone]], singer [[Frank Stallone]], and actress Toni D'Alto, the latter by her former husband Anthony Filiti.

759 Jacqueline Frances Labofish was born on November 29, 1921, in [[Washington, D.C.]],&lt;ref name=&quot;:0&quot; /&gt; the older of two girls. Her father, John Paul Labofish (1891-1956), was a Washington lawyer.{{citation needed¦date=February 2020}} Her mother, Jeanne Victoria Anne &quot;Adrienne&quot; Clerec (1901-1974),&lt;ref&gt;{{cite news ¦url=http://www.letelegramme.com/ig/generales/regions/bretagne/cinema-stallone-est-de-brest-meme-06-10-2009-596723.php ¦title=Stallone est de Brest &quot; même &quot; ! ¦publisher=Le Télégramme de Brest ¦first=Alain ¦last=Coquil ¦date=October 6, 2009¦language=fr}}&lt;/ref&gt; was [[Bretons¦Breton]]. Her parents had met while her father was serving in the [[United States Navy¦U.S. Navy]] at [[Brest (France)¦Brest]] ([[Brittany]]) after the [[First World War]].&lt;ref&gt;{{cite web¦url=http://www.histoire-genealogie.com/spip.php?article2096 ¦title=Jeanne Clerec, la grand-mère de Sylvester Stallone : un destin Américain ¦website=Histoire-Généalogie ¦access-date=October 22, 2014¦language=fr}}&lt;/ref&gt; Her paternal grandparents, Rose (Lamlec) and Charles Labofisz, were Jewish immigrants from [[Odesa]], [[Ukraine]], then a part of the [[Russian Empire]].&lt;ref&gt;{{cite web¦ url=http://omr.gov.ua/en/news/14336/¦ title=Sylvester Stallone's mother visited Odessa¦ website=Official Site of Odessa¦ access-date=July 5, 2015¦ archive-date=January 29, 2019¦ archive-url=https://web.archive.org/web/20190129150204/http://omr.gov.ua/en/news/14336/¦ url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦ url=http://www.artukraine.com/old/historical/stallone.htm¦ title=Mother of movie star Sylvester Stallone seeks grannie in Ukraine¦ website=ArtUkraine.com¦ access-date=July 5, 2015¦ url-status=dead¦ archive-url=https://web.archive.org/web/20140809172653/http://www.artukraine.com/old/historical/stallone.htm¦ archive-date=August 9, 2014}}&lt;/ref&gt;

760 In the 1970s, she owned and operated a facial salon in which she used her own concoctions to treat several skin issues. In her salon, which was on 71st [[Collins Avenue]] in [[Miami Beach]], she had



```
a huge poster of Sylvester Stallone and ''[[The Lords of Flatbush]]''. Stallone was just starting to make it big as an actor and she was very proud. She talked often to her clients about her three children. She enjoyed talking for hours to one client in particular, she would read her palm and entertain her with her love for astrology.
761 Jackie Stallone died on September 21, 2020, at the age of 98. Her death was announced by her son Frank.&lt;ref name=&quot;:0&quot;&gt;{{Cite web|last=Shafer|first=Ellise|date=September 21, 2020|title=Jackie Stallone, Sylvester Stallone's Mother and Celebrity Astrologist, Dies at 98|url=https://variety.com/2020/film/obituaries-people-news/jackie-stallone-dead-sylvester-stallone-mom-1234778583/|access-date=September 22, 2020|website=Variety}}&lt;/ref&gt;&lt;ref&gt;{{cite news|url = https://www.nytimes.com/2020/09/23/style/jackie-stallone-dead.html|title = Jackie Stallone, Celebrity Astrologer and Sylvester's Mother, Dies at 98|work = [[The New York Times]]|date = September 23, 2020|access-date = September 24, 2020|last = Carmel|first = Julia}}&lt;/ref&gt;
762 [[Category:Family of Sylvester Stallone]]
763 | [[Sylvester Stallone]]
764 * [[Sylvester Stallone]] as Lynching Witness (uncredited)
765     <redirect title="Sylvester Stallone" />
766       <text bytes="32" sha1="p7gz58bc0f4qqxa7url52a8fje2gfcq" xml:space="preserve">#REDIRECT [[Sylvester Stallone]]</text>
767 Jay &amp; Silent Bob's Guys We'll Go Gay For in 1993: [[Sylvester Stallone]] and [[Robert Redford]]
768 He was a climbing double for [[Sylvester Stallone]] in the 1993 movie ''[[Cliffhanger (film)|Cliffhanger]]'', along with [[Ron Kauk]].&lt;ref name=&quot;Cliff&quot;&gt;{{cite web|url=https://www.latimes.com/archives/la-xpm-1993-06-02-ca-42466-story.html |title=Making a Movie Out of a Mountain: Climbers Give High Marks to 'Cliffhanger's' Realistic High-Altitude Action Sequences |author=Bill Stall |work=Los Angeles Times |access-date= 13 May 2018}}&lt;/ref&gt;
769 * Film in which Güllich acted as a stunt double for Sylvester Stallone: {{cite AV media | title=Cliffhanger | asin= | type=Motion picture| date=1993 | publisher=RCS Video | people=Renny Harlin | url=https://www.mntnfilm.com/en/film/cliffhanger-1993 | accessdate=2 December 2022}}&lt;ref name=Cliff/&gt;
770 *''[[Rhinestone (film)|Rhinestone]]'' (with [[Sylvester Stallone]]) (1984)
771 Following in his father's footsteps, Zito became an amateur boxer and fought in [[New York Golden Gloves]] while working manual labor. In 1979, after assisting the bodyguards of actor [[Robert Conrad]] at a motorcycle convention at the [[New York Coliseum]], Zito began his own bodyguard agency, Charlie's Angels Bodyguard Service.&lt;ref name=&quot;Muscling into the action&quot;/&gt; Zito initially provided protection for actress [[Lorna Luft]] and later was hired by her half-sister [[Liza Minnelli]].&lt;ref&gt;{{cite news |url=https://query.nytimes.com/gst/fullpage.html?res=9D0DE4DE1F3DF930A1575BC0A96E958260 |title=A Stuntman Is Ready For a Leap to Stardom |date=August 23, 1998 |work=New York Times |access-date=2007-11-01}}&lt;/ref&gt; The actress recommended Zito's service to her plethora of celebrity acquaintances, allowing Zito to quickly develop contacts throughout [[Cinema of the United States|Hollywood]]. His other clients included [[Muhammad Ali]],&lt;ref name=&quot;Friend of the Game&quot;/&gt; [[Charles Bronson]],&lt;ref name=&quot;Muscling into the action&quot;/&gt; [[Michael Jackson]],&lt;ref name=&quot;Muscling into the action&quot;/&gt; [[Sean Penn]],&lt;ref name=&quot;Stuntman Is Ready For Stardom&quot;/&gt; [[Chita Rivera]],&lt;ref name=&quot;Muscling into the action&quot;/&gt; [[Eric Roberts]],&lt;ref name=&quot;Muscling into the action&quot;/&gt; [[Mickey Rourke]],&lt;ref name=&quot;Muscling into the action&quot;/&gt; [[Charlie Sheen]],&lt;ref name=&quot;Who is Zito?&quot;/&gt; and [[Sylvester Stallone]].&lt;ref name=&quot;Muscling into the action&quot;/&gt; Zito's brother-in-law was a policeman, which allowed him to hire moonlighting police officers to bodyguard celebrities at public events which required additional manpower, such as award shows.&lt;ref name=&quot;Muscling into the action&quot;/&gt;
772 Some of the game's enemies are notably too similar to various characters from film and comic books, a fact that Ohba attributes to his own lack of creativity. He made rough sketches of these characters based on what he had in mind at the time, expecting that the character designer would modify them and add their own creative touches for the final game. However, they did not, resulting in some enemy characters who were similar to the likes of [[Sylvester Stallone]], [[Batman]], [[Spider-Man]], [[Terminator (character)|Terminator]] and [[Godzilla]].&lt;ref name=&quot;Retro Gamer&quot;&gt;{{cite news|url=https://www.retrogamer.net/retro_games90/the-making-of-the-revenge-of-shinobi/|title=The Making of the Revenge of Shinobi|date=October 20, 2014|work=[[Retro Gamer]]|issue=96}}&lt;/ref&gt; A Master System version was advertised in a 1990 Sega game catalog, but was never released.&lt;ref&gt;{{cite book |title=Sega For The 90's: The New Generation |url=http://www.smspower.org/Scans/Sega-Catalogue-SegaForThe90s-US?gallerypage=4&amp;sid=61e796488073b1e70c1294f88c483902 |publisher=[[Sega]] |date=1990 |page=4 }}&lt;/ref&gt;
773 Young made his name playing rough-edged working class Italian-American characters, the best-known example being his signature role as Rocky Balboa's friend (and future brother-in-law) Paulie in ''[[Rocky]]'' (1976),&lt;ref&gt;{{cite news |last=Canby |first=Vincent |author-link= Vincent Canby |date=November 22, 1976 |title=Film: 'Rocky,' Pure 30's Make-Believe |url= https://www.nytimes.com/1976/11/22/archives/film-rocky-pure-30s-makebelieve.html |work=The New York Times |access-date=September 23, 2016}}&lt;/ref&gt; for which he received an Oscar nomination for [[Academy Award for Best Supporting Actor|Best Supporting Actor]]. He is one of four actors (the other three being [[Sylvester Stallone]], [[Stu Nahan]], and [[Tony Burton]]) who have appeared in all of the first six [[Rocky (franchise)|''Rocky'' films]] (although [[Talia Shire]], who appears in the first five films, makes a flashback appearance in [[Rocky Balboa (film)|the sixth]]). Young did not reprise his role in the 2015 film ''[[Creed (film)|Creed]]''; the character was described as having died in 2012.&lt;ref&gt;{{cite web |url= http://totalrocky.com/filming-location-guide/rocky-house-creed.html |title= Rocky's House in &quot;Creed&quot; |date= January 19, 2016 |website= Total Rocky |access-date= September 23, 2016 |archive-date= June 1, 2016 |archive-url= https://web.archive.org/web/20160601035714/http://totalrocky.com/filming-location-guide/rocky-house-creed.html |url-status= dead}}&lt;/ref&gt;
774 Young died at [[Northridge Hospital Medical Center]] in Los Angeles, on October 8, 2023,&lt;!--Not 10/18, but 10/8/2023 per cited source--&gt; at the age of 83.&lt;ref&gt;{{cite web |last1=Williams |first1=Alex |title=Burt Young, ''Rocky'' Actor Who Played Complex Tough Guys, Dies at 83 |url=https://www.nytimes.com/2023/10/18/movies/burt-young-dead.html |website=[[The New York Times]] |access-date=October 18, 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://deadline.com/2023/10/sylvester-stallone-burt-young-tribute-rocky-1235577730/|title=Sylvester Stallone Remembers His Late Friend, Burt Young: &quot;I And The World Will Miss You Very Much&quot;|date=19 October 2023 |work=Deadline}}&lt;/ref&gt; The immediate cause of death was ruled as cardiac arrest with contributing factors listed as [[myocardial infarction]], [[atrial fibrillation]], and [[atherosclerosis]]. He was buried at Mount St. Mary Cemetery in [[Flushing, Queens]].&lt;ref&gt;{{cite web|url=https://vt.co/entertainment/celebrity/rocky-burt-youngs-cause-of-death|title='Rocky' star Burt Young's cause of death revealed|first=James|last=Kay|website=VT.co|access-date=March 2, 2024|date=November 9, 2023}}&lt;/ref&gt;
775 Seda appeared in ''[[Treme (TV series)|Treme]]'' seasons 2-4 as Nelson Hidalgo, a politically connected land developer from [[Dallas]] who helps out with the relief efforts in post-[[Hurricane Katrina|Katrina]] [[New Orleans]]. He appeared in the film ''[[Larry Crowne]]'' (2011), which starred and was co-written and directed by [[Tom Hanks]]. Hanks also served as an executive producer on ''The Pacific'', in which Seda starred.&lt;ref&gt;{{cite web|url=https://www.imdb.com/title/tt1583420|title=Larry Crowne (2011)|website=IMDb|date=July 2011}}&lt;/ref&gt; In 2013, Seda starred alongside [[Sylvester Stallone]], [[Christian Slater]], and [[Jason Momoa]] in the [[Walter Hill (director)|Walter Hill]]-directed action thriller ''[[Bullet to the Head]]''.
776 ''Norway'' appears during the closing credits of the 1994 Sylvester Stallone/Sharon Stone feature ''[[The Specialist]]'' in a flyover of the Port of Miami. {{citation needed|date=March 2024}}
777 Crop tops have been worn by men&lt;ref&gt;{{Cite news |last=Payne |first=Marissa |date=August 29, 2015 |title=Ohio State's Ezekiel Elliott wants to trademark 'hero in a half-shirt' |newspaper=Washington Post |url=https://www.washingtonpost.com/news/early-lead/wp/2015/08/29/ohio-states-ezekiel-elliott-wants-to-trademark-hero-in-a-half-shirt/ |url-access=subscription |access-date=29 August 2015}}&lt;/ref&gt; since the 1970s.&lt;ref&gt;{{Cite news |last=Zarley |first=B. David |date=November 7, 2014 |title=Crop Tops In Football: An Investigation |language=en-US |work=VICE Sports |url=https://sports.vice.com/en_us/article/crop-tops-in-football-an-investigation |url-status=dead |access-date=7 November 2014 |archive-url=https://web.archive.org/web/20151211125813/https://sports.vice.com/en_us/article/crop-tops-in-football-an-investigation |archive-date=11 December 2015}}&lt;/ref&gt; The early ''[[Rocky (film series)|Rocky]]'' films have [[Sylvester Stallone]] and [[Carl Weathers]] wearing crop tops while working out.&lt;ref&gt;{{Cite web |date=2023-10-23 |title=6 Types Of Crop Tops That Every Fashionista Should Own |url=https://alyandval.com/2023/10/23/types-of-crop-tops-with-names/ |access-date=2023-12-18 |website=alyandval |language=en-US}}&lt;/ref&gt;
778 As they have for their other songs, Coldplay has refused to accept several offers to use &quot;Trouble&quot; for promotional tools. In 2004, the band rejected a multi-million [[Euro]] offer from [[Diet Coke]] and [[Gap Inc.|Gap]] to use the song and &quot;[[Don't Panic (Coldplay song)|Don't Panic]]&quot;, the fourth single from the album. They asked manager [[Phil Harvey (band manager)|Phil Harvey]] to not refer such offers to them because &quot;a discussion might lead to compromise&quot;. American actor [[Sylvester Stallone]] was interested to use the song for the soundtrack of his 2001 film ''[[Driven (2001 film)|Driven]]'', but the band declined.&lt;ref&gt;{{cite news|url=https://www.theguardian.com/film/2001/aug/16/news1|title=Too many American Psychos|date=16 August 2001|work=[[The Guardian]]|access-date=21 September 2009}}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=http://findarticles.com/p/articles/mi_qn4161/is_/ai_n14526569|title=The Diary: Band in a whole lot of Trouble with Sly|last=Hyland|first=Ian|date=22 April 2001|work=[[Sunday Mirror]] ([[Internet Archive]])|access-date=13 October 2008}}&lt;/ref&gt; The song was used in the season 1 finale of the FX television series ''[[The Shield]]''.
779 Liberal use of rhinestones was associated with country music singers, as well as with singer [[Elvis Presley]] and pianist [[Liberace]]. In 1974 [[David Allan Coe]] released the album ''[[The Mysterious Rhinestone Cowboy]]'' and referred to himself as The Rhinestone Cowboy again in the 1977 song &quot;Longhaired Redneck.&quot; In 1975 [[Glen Campbell]] had a top hit with the song &quot;[[Rhinestone Cowboy]]&quot; and became known as the &quot;Rhinestone Cowboy.&quot; That song served as the basis for the 1984 movie ''[[Rhinestone (film)|Rhinestone]]'' starring [[Sylvester Stallone]] and [[Dolly Parton]]. The British virtual band [[Gorillaz]] released the single &quot;[[Rhinestone Eyes]]&quot; in 2010. The closing track on [[Madvillain]]'s ''[[Madvillainy]]'' is titled &quot;Rhinestone Cowboy.&quot; [[Carrie Underwood]]'s ninth album, released in 2022, is titled &quot;[[Denim &amp; Rhinestones]].&quot;
780 '''''[[Rocky]]''''' is a 1976 film starring Sylvester Stallone.
781 * [[Sylvester Stallone]] (born 1946), American actor
782 * [[Sly (film)|''Sly'' (film)]], a 2023 documentary film about Sylvester Stallone
783 '''''Lovers and Other Strangers''''' is a 1970 American [[romantic comedy]] film directed by [[Cy Howard]], adapted from the 1968 Broadway play of the same name by [[Renée Taylor]] and [[Joseph Bologna]]. The cast includes [[Richard S. Castellano]], [[Gig Young]], [[Cloris Leachman]], [[Anne Jackson]], [[Bea Arthur]], [[Bonnie Bedelia]], [[Michael Brandon]], [[Harry Guardino]], [[Anne Meara]], [[Bob Dishy]], [[Marian Hailey-Moss|Marian Hailey]], Joseph Hindy, and, in her film debut, [[Diane Keaton]]. [[Sylvester Stallone]] was an extra in this movie.&lt;ref name=stallonezone&gt;{{Cite web|url=http://www.stallonezone.com/main/1970/10/sly-scores.html|website = StalloneZone |title= Sly Scores}}&lt;/ref&gt;
784 * [[Sylvester Stallone]] as Groomsman (uncredited)&lt;ref name=stallonezone/&gt;
785 [[Sylvester Stallone]], the director and star of ''Rocky III'', enlisted Survivor to write the song after the band [[Queen (band)|Queen]] denied him permission to use their song &quot;[[Another One Bites the Dust]]&quot;. They derived lyrics, including the title, from dialogue in the film, and conceived a riff with chord changes to match the punches in the boxing scenes.
786 In 1982, the American band [[Survivor (band)|Survivor]] had released two unsuccessful albums, and were concerned they would be dropped by their record label.&lt;ref name=&quot;Grdn&quot;&gt;{{cite news |last1=Jack |first1=Malcolm |date=27 January 2020 |title=Survivor: how we made Eye of the Tiger |work=[[The Guardian]] |publisher= |url=https://www.theguardian.com/music/2020/jan/27/how-we-made-eye-of-the-tiger-rocky-iii-survivor-sylvester-stallone |access-date=20 September 2023}}&lt;/ref&gt; The director and actor [[Sylvester Stallone]] enlisted them to write a song for his 1982 film ''[[Rocky III]]'', after [[Queen (band)|Queen]] denied him permission to use &quot;[[Another One Bites the Dust]]&quot;.&lt;ref name=&quot;Grdn&quot;/&gt;&lt;ref name=&quot;:0&quot;&gt;{{cite magazine |last=Wood |first=Jaes |date=October 19, 2012 |title=Interview: Guitarist Jim Peterik Talks 'Tiger' and the New Pride of Lions Album, 'Immortal' |url=https://www.guitarworld.com/features/interview-guitarist-jim-peterik-talks-tiger-and-new-pride-lions-album-immortal |magazine=[[Guitar World]] |access-date=March 29, 2014}}&lt;/ref&gt; Stallone
```



told them he wanted &quot;something street&quot;&lt;ref name=&quot;Grdn&quot; /&gt; with a pulse to match the punches of the boxing scenes.&lt;ref name=&quot;:1&quot;&gt;{{Cite web |last=Ling |first=Dave |date=2021-07-23 |title=The story behind Survivor's Eye Of The Tiger |url=https://www.loudersound.com/features/the-story-behind-eye-of-the-tiger-by-survivor |access-date=2023-10-01 |website=[[Classic Rock (magazine)|Classic Rock]] |language=en}}&lt;/ref&gt; He sent them a copy of the [[Montage (filmmaking)|montage]] used in the film's introduction, depicting the boxer [[Rocky Balboa]] and the ascent of his rival, [[Clubber Lang]].&lt;ref name=&quot;Grdn&quot; /&gt;&lt;ref name=&quot;:0&quot; /&gt;

787 ''The River of Diamonds'' (1990) had been set for production in the 1960s. During the 1980s [[Brian Clemens]] wrote a script. [[Sylvester Stallone]] - who asked for {{USD|9M}} and a share of the profits - and [[Tom Selleck]] - who asked for {{USD|1.5M}} and a share of the profits - were approached to star, but asked too much money which the production couldn't afford. A journal describes this as South Africa's &quot;most ambitious film project&quot; with what was at the time to have been the biggest budget financed by a South African producer estimated to have been between two and three million [[South African rand|Rand]].&lt;ref name=&quot;SAD&quot;&gt;{{cite journal|journal=South African Digest|year=1984|page=199}}&lt;/ref&gt;

788 One of the main hallmarks of ''How I Spent My Vacation'' was the parody of film, popular culture and celebrities. ''Videos for Kids'' noted that the film makes fun of &quot;California culture and youth&quot; with the use of celebrity caricatures, such as those of [[Roseanne Barr]], [[Johnny Carson]], [[Arsenio Hall]], [[David Letterman]],&lt;ref name = &quot;v4kids&quot;/&gt; [[Jay Leno]], [[Oprah Winfrey]] and [[Sylvester Stallone]].&lt;ref name = &quot;moreceleb&quot;&gt;{{Citation | last = Mink | first = Eric | title = A Looney Vacation With 'Tiny Toons' | newspaper = St. Louis Post-Dispatch | page = 7F | date = March 18, 1992 }}&lt;/ref&gt; The film makes fun of other cartoon characters, such as those of ''[[Superman]]'', ''[[The Simpsons]]'', ''[[Teenage Mutant Ninja Turtles (1987 TV series)|Teenage Mutant Ninja Turtles]]'', ''[[Who Framed Roger Rabbit]]'', ''[[The Ren &amp; Stimpy Show]]'', ''[[Beavis and Butt-Head]]'' and ''[[The Little Mermaid (1989 film)|The Little Mermaid]]''.&lt;ref name = &quot;tvguide&quot;/&gt; Babs and Buster's travels down the river are a parody of the film ''[[Deliverance]]''.&lt;ref name = &quot;firstever&quot;/&gt;

789 *I find it somewhat distressing that &quot;the official most beautiful woman in the world&quot; is compared to [[Sylvester Stallone]], and/or a [[stallion|male horse]]. '''Delete'''! [[User:Samaritan|Samaritan]] 00:19, 7 Feb 2005 (UTC)

790 * 1986 - [[Sylvester Stallone]]

791 In 1985, [[George P. Cosmatos]] made the movie ''[[Rambo: First Blood Part II]]'' (sequel to the 1982 film ''[[First Blood]]'') starring Sylvester Stallone as John Rambo, a Vietnam War veteran who seeks to rescue American prisoners of war left behind in Vietnam whose existence has been covered up.

792 | RTitle = [[Sylvester Stallone]]

793 * '''Backstage:''' The backstage is invaded by Sylvester Stallone's groupies, who have paid Scooter for backstage passes.

794 In ''[[Rocky II]]'', an alternative version of the song was used, with a children's choir singing the chorus. ''[[Rocky III]]'' included an updated disco influenced arrangement during the training montage on the beach. This recording is however missing from the soundtrack album, the sleeve notes of which say &quot;All music on this album selected by [[Sylvester Stallone]]&quot;, who instead opted to reprise the original versions of &quot;Reflections&quot; from the first film, and &quot;Gonna Fly Now&quot; and &quot;Conquest&quot; from the second installment.

795   <text bytes="28898" sha1="efc0nrrclbtouts3bzpzwzeu88kag77" xml:space="preserve">{{short description|1983 film by Sylvester Stallone}}

796 | director   = [[Sylvester Stallone]]

797 * Sylvester Stallone

798 * Sylvester Stallone

799 | released   = {{Film date|1983|7|11|[[Grauman's Chinese Theatre|Mann's Chinese Theatre]]|ref1=&lt;ref&gt;{{cite web |url=https://catalog.afi.com/Catalog/moviedetails/58134 |title=Staying Alive |author=&lt;!--Not stated--&gt; |website=[[AFI Catalog of Feature Films]] |publisher=[[American Film Institute]] |access-date=2018-06-11 |quote=...&amp;nbsp;the 22 Jun 1983 Var announced premiere events in Los Angeles at the Chinese Theatre on 11 Jul 1983, and in New York City at the Ziegfeld Theater on 13 Jul 1983&amp;nbsp;... |archive-date=12 June 2018 |archive-url=https://web.archive.org/web/20180612141046/https://catalog.afi.com/Catalog/moviedetails/58134 |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{Cite AV media notes |title=John Travolta and Sylvester Stallone during 'Stayin' Alive' Premiere |date=1983-07-11 |url=https://www.gettyimages.com/license/115412569 |first=Ron |last=Galella |author-link=Ron Galella |type=photography |publisher=[[Getty Images]] |id=115412569 |location=Seattle |editor-last=Smeal|editor-first=Jim}}&lt;/ref&gt;|1983|7|15|United States}}

800 ''''''Staying Alive'''''' is a 1983 American [[dance in film|dance]] [[drama film]] and the [[sequel]] to ''[[Saturday Night Fever]]'' (1977). The film was directed by [[Sylvester Stallone]], who co-produced and co-wrote the film with original ''Fever'' producer [[Robert Stigwood]], and writer [[Norman Wexler]]. ''Staying Alive'' stars [[John Travolta]], reprising his ''Saturday Night Fever'' role as Tony Manero, with [[Cynthia Rhodes]], [[Finola Hughes]], [[Joyce Hyser]], [[Julie Bovasso]], [[Viktor Manoel]] and [[Kevyn Morrow]].

801 ''Staying Alive'' was theatrically released on July 15, 1983, to universally negative critical reviews, and is the oldest film to hold a [[List of films with a 0% rating on Rotten Tomatoes|score of 0%]] on [[Rotten Tomatoes]]. Despite this, ''Staying Alive'' was successful at the box office, earning $127 million worldwide on a $22 million budget. The film also featured the song &quot;[[Far from Over (Frank Stallone song)|Far from Over]]&quot; by [[Frank Stallone]], the younger brother of [[Sylvester Stallone]]. &quot;Far from Over&quot; peaked at #10 on the [[Billboard Hot 100|''Billboard'' Hot 100]] and the ''[[Cashbox (magazine)|Cashbox]]'' charts. Along with ''[[Homefront (2013 film)|Homefront]]'' (2013), this is one of only two films that Stallone wrote without starring (although he has an uncredited [[cameo appearance]]).

802 * [[Sylvester Stallone]] as Man on Street (uncredited)

803 [[Richie Sambora]] appeared in an uncredited role as a guitarist of the local band, in which Jackie and Carl also perform.&lt;ref&gt;{{cite news |last=Wuench |first=Kevin |date=May 18, 2017 |title=The Bee Gees fared as well in '80s as the movie 'Staying Alive' |url=https://www.tampabay.com/the-bee-gees-fared-as-well-in-80s-as-the-movie-staying-alive/2324407/ |work=[[Tampa Bay Times]] |location=[[St. Petersburg, Florida|St. Petersburg]] |quote=If you look real close, you can catch Richie Sambora (Bon Jovi) in Frank Stallone's on-film band&amp;nbsp;... |access-date=February 15, 2021 |archive-date=April 22, 2023 |archive-url=https://web.archive.org/web/20230422101412/https://www.tampabay.com/the-bee-gees-fared-as-well-in-80s-as-the-movie-staying-alive/2324407/ |url-status=live }}&lt;/ref&gt; Sylvester Stallone makes an uncredited [[cameo appearance]] as a man on the street, whom Tony bumps into.

804 The soundtrack album was released in 1983 and is performed mostly by the [[Bee Gees]]. Five new Bee Gees songs were on Side one, with Side two featuring songs by other artists which were mostly written by [[Frank Stallone]], brother of the film's director, [[Sylvester Stallone]]. The album reached No. 14 in the United Kingdom, No. 6 in the United States, No. 1 in Switzerland, and No. 2 in Italy and Japan. The Bee Gees songs were released under [[RSO Records|RSO]].

805 {{Sylvester Stallone}}

806 [[Category:Films directed by Sylvester Stallone]]

807 [[Category:Films with screenplays by Sylvester Stallone]]

808 | style=&quot;background:#B0C4DE;&quot; | '''[[Sylvester Stallone]] ''' || style=&quot;background:#B0C4DE;&quot; | '''[[Rocky Balboa]]''' || style=&quot;background:#B0C4DE;&quot; | '''''[[Creed (film)|Creed]]'''''

809 | ''[[Rocky]]'' || [[Sylvester Stallone]]

810 In 2017, Metcalfe starred as Eric Redford in the Hallmark movie ''[[Christmas Next Door]]'' alongside [[Fiona Gubelmann]].&lt;ref&gt;{{Cite web|url=http://www.hallmarkchannel.com/christmas-next-door/cast|title='''Christmas Next Door''' Cast|website=Hallmark Channel|access-date=December 19, 2017}}&lt;/ref&gt; The film premiered to more than 4.4 million viewers, becoming top 10 cable program.&lt;ref&gt;{{cite web |url=http://www.showbuzzdaily.com/articles/showbuzzdailys-top-150-saturday-cable-originals-network-finals-12-16-2017.html |archive-url=https://web.archive.org/web/20171219224738/http://www.showbuzzdaily.com/articles/showbuzzdailys-top-150-saturday-cable-originals-network-finals-12-16-2017.html |url-status=dead |archive-date=December 19, 2017 |title=SHOWBUZZDAILY's Top 150 Saturday Cable Originals &amp; Network Finals: 12.16.2017|work=ShowBuzzDaily |first=Mitch |last=Fratti |date=December 16, 2017 |access-date=February 12, 2020}}&lt;/ref&gt; In 2018, Metcalfe played the role of Luke in the action thriller ''[[Escape Plan 2: Hades]]''.&lt;ref&gt;{{Cite news|url=http://www.tracking-board.com/jesse-metcalfe-pete-wentz-join-sylvester-stallone-in-escape-plan-2/|title=Jesse Metcalfe, Pete Wentz Join Sylvester Stallone in &quot;Escape Plan 2&quot;|date=March 21, 2017|work=The Tracking Board|access-date=July 26, 2018|language=en-US}}&lt;/ref&gt; He also starred as Brady in the horror thriller film ''[[The Ninth Passenger]]''.&lt;ref&gt;{{Cite news|url=https://variety.com/2016/film/news/lionsgate-jesse-metcalfe-the-ninth-passenger-1201810393/|title=Lionsgate UK in Talks to Distribute Jesse Metcalfe's 'The Ninth Passenger'|last=McNary|first=Dave|date=July 8, 2016|work=Variety|access-date=July 26, 2018|language=en-US}}&lt;/ref&gt; In 2019, Metcalfe starred in the Hallmark channel Christmas film ''Christmas Under the Stars'' co-starring [[Autumn Reeser]].&lt;ref&gt;{{Cite web|url=https://www.hallmarkchannel.com/christmas-under-the-stars|title=Christmas Under the Stars|website=Hallmark Channel|language=en|access-date=November 19, 2019}}&lt;/ref&gt;

811 * [[Sylvester Stallone]], actor

812 [[Category:Images of Sylvester Stallone]]</text>

813 Kercheval gained his first television role, playing the part of Dr. Nick Hunter number one on ''[[Search for Tomorrow]]'' in 1966. His later soap-opera roles were in ''[[The Secret Storm]]'' and ''[[How to Survive a Marriage]]''. His film credits include ''[[The Seven-Ups]]'' with [[Roy Scheider]] and [[Tony LoBianco]] plus ''[[F.I.S.T. (film)|F.I.S.T.]]'' with [[Sylvester Stallone]]. In 1976, he co-starred in two episodes of ''[[The Adams Chronicles]]'' as [[James Madison]].

814 Wright's subsequent releases focused on film soundtracks and forays into [[world music]].&lt;ref name=&quot;Wright bio/AM&quot; /&gt; After writing the score for [[Alan Rudolph]]'s 1982 thriller ''[[Endangered Species (1982 film)|Endangered Species]]'',&lt;ref name=&quot;DreamWeaverBio&quot;&gt;{{cite web|title=Gary Wright &gt; Biography|url=http://www.thedreamweaver.com|publisher=thedreamweaver.com|access-date=March 21, 2014}}&lt;/ref&gt; he supplied the soundtrack to another skiing-themed movie by Willy Bogner,&lt;ref&gt;{{cite web|title=Willy Bogner Film: Action Videos &amp; Sport Movies|url=http://en.bogner.com/Discover/Willy-Bogner-Film|publisher=bogner.com|access-date=March 21, 2014|url-status=dead|archive-url=https://web.archive.org/web/20140401162616/http://en.bogner.com/Discover/Willy-Bogner-Film|archive-date=April 1, 2014}}&lt;/ref&gt; ''[[Fire and Ice (1986 film)|Fire and Ice]]'' (1986), which hit number 1 on the German albums chart.&lt;ref name=&quot;Wright bio/AM&quot; /&gt; Wright also contributed the song &quot;Hold on to Your Vision&quot; to the soundtrack of ''[[Cobra (1986 film)|Cobra]]'', a 1986 [[Action film|action movie]] starring [[Sylvester Stallone]] as well as a [[pop rock]] song &quot;Am I the One&quot; for the 1987 [[Surf film|surf movie]], ''[[North Shore (1987 film)|North Shore]]''.&lt;ref name=&quot;IMDB/North Shore&quot;&gt;{{cite web|title=IMDB North Shore soundtrack| website=[[IMDb]] |url=https://www.imdb.com/title/tt0093648/soundtrack/?ref_=tt_trv_snd}}&lt;/ref&gt;

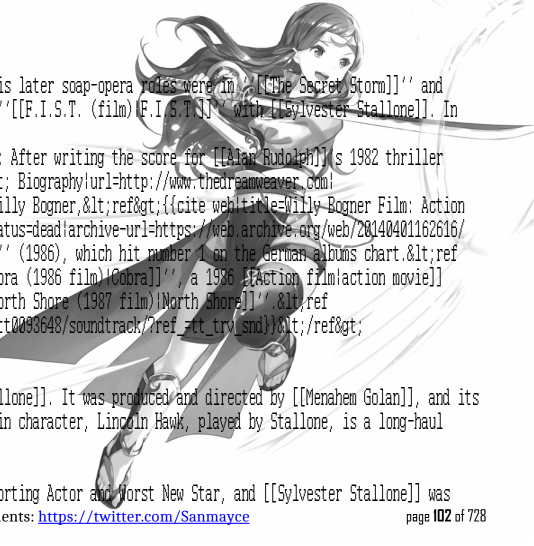
815 * [[Sylvester Stallone]]

816 * Sylvester Stallone

817 ''''''Over the Top'''''' is a 1987 American [[sports film|sports]] [[Drama (film and television)|drama film]] starring [[Sylvester Stallone]]. It was produced and directed by [[Menahem Golan]], and its screenplay was written by [[Stirling Silliphant]] and Stallone. The original music score was composed by [[Giorgio Moroder]]. The main character, Lincoln Hawk, played by Stallone, is a long-haul truck driver who tries to win back his estranged son, Michael, while becoming a champion [[arm wrestling|arm wrestler]].

818 * [[Sylvester Stallone]] as Lincoln &quot;Linc&quot; Hawk

819 The film received three nominations at the [[8th Golden Raspberry Awards]] in 1988. [[David Mendenhall]] won two for both Worst Supporting Actor and Worst New Star, and [[Sylvester Stallone]] was



nominated for Worst Actor, which he lost to [[Bill Cosby]] for ''[[Leonard Part 6]]''.

820 {{Sylvester Stallone}}

821 [[Category:Films with screenplays by Sylvester Stallone]]

822 * The film ''[[Over the Top (1987 film)|Over the Top]]'' (1987) features an arm wrestling champion, played by [[Sylvester Stallone]].

823 [[Sylvester Stallone]] played Stakar in the [[Marvel Cinematic Universe]]&lt;ref name=&quot;screencrush.com&quot;&gt;{{cite web|url=http://screencrush.com/kurt-russell-and-sylvester-stallone-will-return-for-more-marvel-movies/|title=Russell and Stallone Will Make Marvel Movies After 'GotG2'|website=ScreenCrush|date=March 25, 2017 |access-date=15 May 2017}}&lt;/ref&gt; films ''[[Guardians of the Galaxy Vol. 2]]'' (2017) and ''[[Guardians of the Galaxy Vol. 3]]'' (2023).

824 [[File:Sylvester Stallone as Stakar Ogord.jpg|right|thumb|Sylvester Stallone as Stakar Ogord in ''[[Guardians of the Galaxy Vol. 2]]'']]

825 * Stakar Ogord / Starhawk appears in the [[Marvel Cinematic Universe]] (MCU) film ''[[Guardians of the Galaxy Vol. 2]]'', portrayed by [[Sylvester Stallone]].&lt;ref name=&quot;screencrush.com&quot;/&gt; This version is the leader of a faction of [[Ravagers (Marvel Cinematic Universe)|Ravagers]] and is acquainted with [[Yondu Udonta (Marvel Cinematic Universe)|Yondu Udonta]],&lt;ref&gt;{{cite web|url=http://comicbook.com/marvel/2017/04/17/guardians-of-the-galaxy-vol-2-sylvester-stallone-role/|title=Sylvester Stallone's Mystery Role In Guardians Of The Galaxy Vol. 2 Revealed|last=Peters|first=Megan|work=ComicBook.com|date=April 17, 2017|archive-url=https://web.archive.org/web/20170418233935/http://comicbook.com/marvel/2017/04/17/guardians-of-the-galaxy-vol-2-sylvester-stallone-role/|archive-date=April 18, 2017|url-status=live}}&lt;/ref&gt; having saved him from being a [[Kree]] slave. Stakar and his second [[Martinex (Marvel Cinematic Universe)|Martinex T'Naga]] run into Yondu on the planet Contraxia, where they remind him he is in exile for breaking the Ravagers' code of never trafficking children. After learning Yondu had sacrificed himself to save [[Peter Quill (Marvel Cinematic Universe)|Peter Quill]], the Ravagers attend the former's funeral in his honor.&lt;ref&gt;{{cite web|author=Gaudette, Emily|url=https://www.inverse.com/article/31172-guardians-of-the-galaxy-vol-2-yondu-ravagers-sylvester-stallone|publisher=Inverse|title=Why the Ravagers Reject Yondu in 'Guardians Vol. 2'|date=May 9, 2017|access-date=November 11, 2019|archive-date=July 24, 2019|archive-url=https://web.archive.org/web/20190724202100/https://www.inverse.com/article/31172-guardians-of-the-galaxy-vol-2-yondu-ravagers-sylvester-stallone|language=en}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.marvel.com/characters/star-lord-peter-quill/on-screen|publisher=[[Marvel Entertainment]]|title=PETER QUILL STAR-LORD|date=2018|access-date=November 12, 2019|archive-date=November 12, 2019|archive-url=https://archive.today/20191112021114/https://www.marvel.com/characters/star-lord-peter-quill/on-screen|language=en|url-status=live}}&lt;/ref&gt; In a mid-credits scene, Stakar and Martinex reunite with their old teammates [[Charlie-27]], [[Aleta Ogord (Marvel Cinematic Universe)|Aleta Ogord]], [[Mainframe (comics)|Mainframe]], and [[Krugarr]] in Yondu's memory.

826 * Stakar Ogord appears in the MCU film ''[[Guardians of the Galaxy Vol. 3]]'', portrayed again by Sylvester Stallone.&lt;ref&gt;{{cite web |last1=DeVore |first1=Britta |title=Sylvester Stallone Confirms His Return to the MCU in 'Guardians of the Galaxy Vol. 3' |url=https://collider.com/guardians-of-the-galaxy-3-sylvester-stallone/ |website=[[Collider (website)|Collider]] |access-date=12 November 2021 |date=November 12, 2021}}&lt;/ref&gt; Stakar and his clan of Ravagers assisted the Guardians of the Galaxy in breaking into the [[High Evolutionary]]'s Orgosphere.

827 Mangold subsequently wrote and directed ''[[Cop Land]]'' (1997), starring [[Sylvester Stallone]], [[Robert De Niro]], [[Harvey Keitel]], and [[Ray Liotta]];&lt;ref&gt;{{cite news|last=Blair|first=Ian|url=https://news.google.com/newspapers?nid=1309&amp;dat=19970821&amp;id=5v1OAAAAIBAJ&amp;pg=5953,286323 |title=Mangold gets big names to come on board|work=[[New Straits Times]]|date=August 21, 1997|page=5|access-date=July 22, 2013}}&lt;/ref&gt; ''[[Girl, Interrupted (film)|Girl, Interrupted]]'',&lt;ref name=&quot;Ivry 00&quot;&gt;{{cite news|last=Ivry|first=Bob|url=https://news.google.com/newspapers?nid=1955&amp;dat=20000111&amp;id=DYs0AAAAIBAJ&amp;pg=4204,6160375 |title=Director brought edge to 'chick flick'|work=[[Reading Eagle]]|date=January 11, 2000|page=D5|access-date=July 22, 2013}}&lt;/ref&gt; which won the [[Academy Award for Best Supporting Actress]] in 1999 for [[Angelina Jolie]]; ''[[Kate &amp; Leopold]]'',&lt;ref&gt;{{cite news|last=Turner|first=Miki|url=https://news.google.com/newspapers?nid=1346&amp;dat=20011226&amp;id=jtxOAAAAIBAJ&amp;pg=6377,3897235 |title=Hugh Jackman's Charm|work=[[Lakeland Ledger]]|date=December 26, 2001|page=D5|access-date=July 22, 2013}}&lt;/ref&gt; starring [[Meg Ryan]] and [[Hugh Jackman]], for which Jackman was nominated for a Golden Globe as best actor in a musical or comedy in 2001, and the 2003 thriller ''[[Identity (2003 film)|Identity]]'' which starred [[John Cusack]].&lt;ref&gt;{{cite news|last=Tinsley|first=Brad|url=https://news.google.com/newspapers?nid=1876&amp;dat=20030912&amp;id=UQooAAAAIBAJ&amp;pg=3900,4343753 |title='Identity' is a hair-raiser; 'Anger Management' frustrates (Review)|work=[[Spartanburg Herald-Journal]]|date=September 12, 2003|page=D7|access-date=July 22, 2013}}&lt;/ref&gt;

828 Dickinson has been married four times. Her former husbands are Ron Levy,&lt;ref name=usmagazine.com /&gt;&lt;ref&gt;{{cite web|url=http://m.zimbio.com/Ron+Levy+Dating+History/articles/V8URjZmGiEv/Ron+Levy+married+Janice+Dickinson|title=Ron Levy was married to Janice Dickinson - Ron Levy Dating History|publisher=Zimbio.com|access-date=April 7, 2016}}&lt;/ref&gt; Simon Fields,&lt;ref name=usmagazine.com /&gt; and Alan B. Gersten,&lt;ref name=usmagazine.com /&gt; also known as Albert Gersten.&lt;ref name=nndb /&gt; She has a son, Nathan, and a daughter, Savannah.&lt;ref name=usmagazine.com /&gt; Dickinson was having an affair with [[Sylvester Stallone]] when Savannah was born in 1994, and it was reported that Stallone was the father.&lt;ref&gt;{{Cite web|url=https://people.com/archive/rocky-ending-vol-41-no-16/|title=Rocky Ending|last=Schneide|first=Karen S.|date=March 2, 1994|website=People}}&lt;/ref&gt; Their relationship ended when DNA tests proved he was not the father.&lt;ref&gt;{{Cite web|url=https://people.com/archive/yo-angie-marry-me-vol-43-no-16/|title=Yo, Angie, Marry Me!|last=Levitt|first=Shelley|date=April 24, 1995|website=People}}&lt;/ref&gt; In her books and in interviews, she has discussed her numerous sexual relationships with male and female celebrities.&lt;ref&gt;{{cite web|url=http://www.nydailynews.com/entertainment/story/194022p-167656c.html |archive-url=https://web.archive.org/web/20040528121419/http://www.nydailynews.com/entertainment/story/194022p-167656c.html|url-status=dead|archive-date=May 28, 2004|title=Dickinson, on the Covers - and Under Them |last=MacKenzie|first=Drew|work=[[Daily News (New York)|Daily News]]|location=New York City|access-date=September 24, 2006}}&lt;/ref&gt; In 2012, she announced she was engaged to Dr. Robert Gerner (&quot;Rocky&quot;),&lt;ref name=us-engagement&gt;{{cite news|url=http://www.usmagazine.com/celebrity-news/news/janice-dickinson-couldnt-be-happier-over-engagement-to-dr-robert-gerner-20121612|title=Janice Dickinson &quot;Couldn't Be Happier&quot; Over Engagement to Dr. Robert Gerner|work=[[Us Weekly]]|date=December 16, 2012|access-date=March 28, 2016}}&lt;/ref&gt; a psychiatrist&lt;ref&gt;[http://www.healthgrades.com/physician/dr-robert-gerner-2sph3 Dr. Robert Gerner profile], healthgrades.com; accessed November 19, 2014.&lt;/ref&gt; whom she married in December 2016.&lt;ref&gt;{{cite web|url=https://www.etonline.com/news/204726_janice_dickinson_is_married|date=December 10, 2006|access-date=May 15, 2019|work=ET Online|title=Janice Dickinson Marries Dr. Robert Gerner in Beverly Hills Ceremony}}&lt;/ref&gt;

829 * {{anchor|Sylvester Stallone}}''''[[Sylvester Stallone]]''': The actor was subject to an Internet death hoax on September 3, 2016.&lt;ref&gt;Hood, Micaela (September 5, 2016). [http://www.nydailynews.com/entertainment/sylvester-stallone-victim-celebrity-death-hoax-article-1.2779081 Sylvester Stallone victim of celebrity death hoax]. ''New York Daily News''. Retrieved September 7, 2016.&lt;/ref&gt; In February 2018, with another hoax circulating, he posted to Instagram that he was alive.&lt;ref&gt;{{citation|title=Sylvester Stallone is alive and 'still punching' despite death hoax|magazine=Entertainment Weekly|date=February 19, 2018|url=https://ew.com/news/2018/02/19/sylvester-stallone-alive-death-hoax/|access-date=February 20, 2020|archive-date=August 16, 2019|archive-url=https://web.archive.org/web/20190816054628/https://ew.com/news/2018/02/19/sylvester-stallone-alive-death-hoax/|url-status=dead}}&lt;/ref&gt;

830 Upon hearing that [[Tolkien Black|Token]] has won a contest that will allow him to sing at a [[Colorado]] beauty pageant and receive $200, [[Stan Marsh|Stan]], [[Eric Cartman|Cartman]], [[Kyle Broflovski|Kyle]] and [[Kenny McCormick|Kenny]] decide to set up the &quot;Super Awesome Talent Agency&quot; and obtain 10% of his earnings by becoming his agents. They lose him to [[Creative Artists Agency]], however, only to land a singer named [[Wing (singer)|Wing]], who is the wife of the City Wok owner, [[Tuong Lu Kim]]. Recently smuggled into the [[United States]] by the [[Triad (underground society)|Triad]]s, Wing has been set to audition for ''[[American Idol]]'', and the boys agree to bring her to [[Los Angeles]] for the competition. This venture does not go as planned, and the boys instead enter her into ''[[The Contender (TV series)|The Contender]]'', a television series about [[boxing]]. [[Sylvester Stallone]] is impressed with her singing, even as she is beaten, and gives her a chance to sing at his son's wedding, which will give the boys a 10% share of $4000.

831 [[Category:Cultural depictions of Sylvester Stallone]]

832 In 1959, McLendon co-produced and co-starred in two sci-fi monster movies filmed in Texas, ''[[The Killer Shrews]]'' and ''[[The Giant Gila Monster]]''. He produced over 150 motion-picture campaigns for [[United Artists]] from 1963 to 1966. At one point, he became the largest shareholder in [[Columbia Pictures]]. He was the executive producer of ''[[Escape to Victory]]'', directed by [[John Huston]] and starring [[Michael Caine]], [[Sylvester Stallone]], and [[Max von Sydow]]. He also owned McLendon Theatres, which operated more than forty movie theatres throughout the south, including many [[drive-in theater|drive-ins]].&lt;ref&gt;{{cite web|title=Movie Theaters Previously Operated by McLendon Theatres|url=http://cinematreasures.org/chains/786/previous?status=all|website=Cinema Treasures|access-date=March 13, 2018}}&lt;/ref&gt;

833 Director John Herzfeld explained in an interview with [[Sylvester Stallone]] that the original cut of the film was considered to be too controversial and included [[Orson Welles]] as God. The interview is a special feature on the Blu-ray for Herzfeld's ''[[2 Days in the Valley]]'' released by [[Kino Lorber]]. Herzfeld said the original cut was never screened for test audiences and upon studio alterations, [[Gene Hackman]] was brought in to replace Welles and record new audio for the re-edited film.

834 * [[Sylvester Stallone]] as Youth in Park (suspected pickpocket)

835 The film version of ''The Prisoner of Second Avenue'' stars [[Jack Lemmon]], [[Anne Bancroft]] and [[Gene Saks]]. It was produced and directed by [[Melvin Frank]] from a screenplay by Simon. The music is by [[Marvin Hamlisch]].&lt;ref name=weiler&gt;Weiler, A. H. (March 15, 1975). [https://www.nytimes.com/1975/03/15/archives/film-a-new-neil-simonprisoner-of-second-avenue-opens.html Film: A New Neil Simon&quot;]. ''[[The New York Times]]''. p. 18.&lt;/ref&gt; [[Sylvester Stallone]] appears in a brief role as a suspected mugger of Jack Lemmon's character.

836 * [[Sylvester Stallone]], actor

837 * [[Cliffhanger (film)|''Cliffhanger'' (film)]], 1993 action film starring Sylvester Stallone and John Lithgow

838 * [[Sylvester Stallone]]

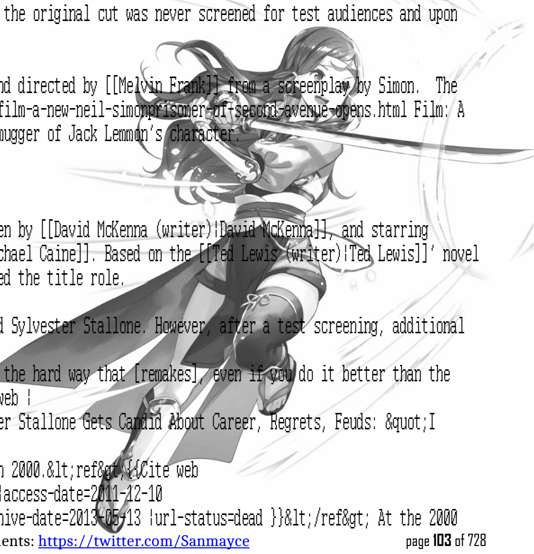
839 '''''Get Carter''''' is a 2000 American [[action film|action]] [[thriller (genre)|thriller film]] directed by [[Stephen Kay]], written by [[David McKenna (writer)|David McKenna]], and starring [[Sylvester Stallone]], [[Rachael Leigh Cook]], [[Alan Cumming]], [[Mickey Rourke]], [[John C. McGinley]], [[Rhona Mitra]], and [[Michael Caine]]. Based on the [[Ted Lewis (writer)|Ted Lewis]]' novel ''[[Jack's Return Home]]'', the film is a [[remake]] of the [[Get Carter|1971 film of the same name]], in which a younger Caine played the title role.

840 * [[Sylvester Stallone]] as Jack Carter

841 [[Michael Caine]]'s role was originally relegated to a one-scene [[cameo appearance]], which he agreed to do as a favor to his friend Sylvester Stallone. However, after a test screening, additional scenes were scripted and shot to expand this role.

842 Later Stallone said: &quot;Believe it or not, I think ''Get Carter'' was really underrated. That was a big disappointment. I learned the hard way that [remakes], even if you do it better than the original, there's a tremendous nostalgia attached to the original. And quite often they're not done as well.&quot;&lt;ref&gt;{{cite web | url=https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/ | title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot; | website=[[The Hollywood Reporter]] | date=7 November 2022 }}&lt;/ref&gt;

843 The film was nominated for Worst Actor ([[Sylvester Stallone]]) and Worst Remake or Sequel at the [[21st Golden Raspberry Awards]] in 2000.&lt;ref&gt;{{Cite web |url=http://www.razzies.com/forum/2000-razzie-nominees-winners_topic350.html |title=2000 RAZZIE® Nominees &amp; &quot;Winners&quot; |access-date=2011-12-10 |archive-url=https://web.archive.org/web/20130513120507/http://www.razzies.com/forum/2000-razzie-nominees-winners_topic350.html |archive-date=2013-05-13 |url-status=dead }}&lt;/ref&gt; At the 2000



[[Stinkers Bad Movie Awards]], the film received four nominations: Worst Picture, Worst Director (Kay), Worst Actor (Stallone), and Worst Remake/Sequel.&lt;ref&gt;{{cite web |title=Past Winners Database |url=http://theenvelope.latimes.com/extras/lostmind/year/2000/2000st.htm |archive-url=https://web.archive.org/web/20070105221207/http://theenvelope.latimes.com/extras/lostmind/year/2000/2000st.htm |url-status=dead |archive-date=5 January 2007 |website=The Envelope at LA Times |access-date=18 September 2019}}&lt;/ref&gt;

844 | Truck driver Lincoln &quot;Linc&quot; Hawk ([[Sylvester Stallone]]) enters an arm-wrestling competition.

845 In 1993 Sylvester Stallone was attached to the project.&lt;ref&gt;{{cite web |date=1 June 1993 |last=Moerk |first=Christian |title=Silver signs Stallone for 'Game' |url=https://variety.com/1993/film/news/silver-signs-stallone-for-game-107327/ |website=[[Variety (magazine)|Variety]] }}&lt;/ref&gt; [[Geena Davis]], [[Julianne Moore]] and [[Brooke Shields]] were all offered the role of Kate McQuean, but they all passed as they were busy with other projects, before supermodel [[Cindy Crawford]] was ultimately cast.

846 * [[Spiro Razatos]], who directed second unit on ''[[Bad Boys II]]'' &amp; ''[[Transformers: The Last Knight]]'' for Michael Bay, ''[[Bad Boys For Life]]'' for [[Adil El Arbi and Bilall Fallah]], ''[[Talladega Nights: The Ballad of Ricky Bobby]]'' for Adam McKay, ''[[Kong: Skull Island]]'' for [[Jordan Vogt-Roberts]], ''[[X-Men: First Class]]'' for Matthew Vaughn, ''[[The Expendables (2010 film)|The Expendables]]'' for [[Sylvester Stallone]], ''[[Fast Five]]'', ''[[Fast &amp; Furious 6]]'', &amp; ''[[F9: The Fast Saga]]'' for Justin Lin, ''[[Furious 7]]'' for [[James Wan]], ''[[The Fate of the Furious]]'' for F. Gary Gray, ''[[Fast X]]'' for [[Louis Leterrier]], ''[[Captain America: The Winter Soldier]]'', ''[[Captain America: Civil War]]'', &amp; ''[[The Gray Man (2022 film)|The Gray Man]]'' for Anthony and Joe Russo, ''[[Venom (2018 film)|Venom]]'' for [[Ruben Fleischer]], ''[[Death Race (2008 film)|Death Race]]'' for [[Paul W.S. Anderson]], ''[[Total Recall (2012 film)|Total Recall]]'' for Len Wiseman, ''[[Driven (2001 film)|Driven]]'' for [[Renny Harlin]], and ''[[21 Grams]]'' for [[Alejandro Gonzalez Inarritu]].

847 * [[Terry J. Leonard]], who did ''[[The Expendables (2010 film)|The Expendables]]'' for Sylvester Stallone, ''[[Rush Hour (1998 film)|Rush Hour]]'' for Brett Ratner, ''[[Cowboys &amp; Aliens]]'' for [[Jon Favreau]], ''[[Gangster Squad (film)|Gangster Squad]]'' for Ruben Fleischer, ''[[Die Hard With a Vengeance]]'' for John McTiernan, ''[[Used Cars]]'' &amp; ''[[Romancing the Stone]]'' for [[Robert Zemeckis]], ''[[2 Fast 2 Furious]]'' for [[John Singleton]], ''[[The Fast and the Furious: Tokyo Drift]]'' for Justin Lin, ''[[Teenage Mutant Ninja Turtles II: The Secret of the Ooze]]'' for [[Michael Pressman]], ''[[American Outlaws (2001 film)|American Outlaws]]'' for [[Les Mayfield]], ''[[Joy Ride (2001 film)|Joy Ride]]'' for [[John Dahl]], ''[[Tombstone (film)|Tombstone]]'' for George P. Cosmatos, ''[[The Quick and the Dead (1995 film)|The Quick and The Dead]]'' for Sam Raimi, ''[[Maverick (film)|Maverick]]'' for Richard Donner, ''[[The Fugitive (1993 film)|The Fugitive]]'' &amp; ''[[The Package (1989 film)|The Package]]'' for Andrew Davis, ''[[Eraser (film)|Eraser]]'' for [[Chuck Russell]], ''[[Flight of the Intruder]]'', ''[[Big Wednesday]]'', &amp; ''[[Conan the Barbarian (1982 film)|Conan the Barbarian]]'' for [[John Milius]], and ''[[Date Movie]]'' for [[Jason Friedberg and Aaron Seltzer]].

848 * Thomas J. Wright, who did ''[[Staying Alive (1983 film)|Staying Alive]]'' for Sylvester Stallone, ''[[Howard the Duck (film)|Howard the Duck]]'' for [[Willard Huyck]], ''[[Neil Simon's I Ought to Be in Pictures]]'' &amp; ''[[Pennies from Heaven (1981 film)|Pennies from Heaven]]'' for [[Herbert Ross]], ''[[Beverly Hills Cop]]'' for [[Martin Brest]], and ''[[Final Destination (film)|Final Destination]]'' for [[James Wong (filmmaker)|James Wong]].

849 * Harvey Harrison, who did ''[[Rambo (2008 film)|Rambo]]'' for Sylvester Stallone, ''[[The Avengers (1998 film)|The Avengers]]'' for Jeremiah S. Chechik, ''[[Around the World in 80 Days (2004 film)|Around the World in 80 Days]]'' for Frank Coraci, ''[[Equilibrium (film)|Equilibrium]]'' for [[Kurt Wimmer]], ''[[The Nativity Story]]'' for Catherine Hardwicke, and ''[[V For Vendetta (film)|V For Vendetta]]'' for [[James McTeigue]].

850 * [[Charles Winkler]], who directed second unit on ''[[Rocky Balboa (film)|Rocky Balboa]]'' for Sylvester Stallone (in additional to serving as producer) and ''[[Home of the Brave (2006 film)|Home of the Brave]]'' for his father [[Irwin Winkler]].

851 By 1995 Phillips was a multi-millionaire, and was well known in celebrity and sports circles. Athletes like [[José Canseco]] would contact Phillips for advice on steroids,&lt;ref&gt;[http://www.t-nation.com/free_online_article/sex_news_sports_funny_social/atomic_dog_roid_ball;jsessionid=E45987F8B10B4E469DD997703ACA083E.hydra Atomic Dog 'Roid Ball by TC]&lt;/ref&gt; and he also consulted with celebrities such as [[Jerry Seinfeld]], [[John Elway]], [[Sylvester Stallone]] and [[Demi Moore]].&lt;ref name=&quot;coloradorunnermag.com&quot;&gt;{{Cite web |url=http://www.coloradorunnermag.com/read_new/body-life-author-bill-phillips-partners-denver-marathon-official-sponsor |title=Body For Life Author Bill Phillips Partners with Denver Marathon as Official Sponsor |access-date=2008-11-25 |archive-url=https://web.archive.org/web/20100330081143/http://www.coloradorunnermag.com/read_new/body-life-author-bill-phillips-partners-denver-marathon-official-sponsor |archive-date=2010-03-30 |url-status=dead }}&lt;/ref&gt;&lt;ref&gt;[http://www.ergogenics.org/231.html Abbott Labs to buy EAS Kristi Arellano, The Denver Post, October 12, 2004]&lt;/ref&gt;

852 Brambilla made his directorial debut with the [[futuristic]] action film ''[[Demolition Man (film)|Demolition Man]],'' premiered October 8, 1993, starring [[Sylvester Stallone]], [[Wesley Snipes]] and [[Sandra Bullock]].&lt;ref&gt;{{Cite web|title=AFI{{!}}Catalog|url=https://catalog.afi.com/Film/67651-DEMOLITION-MAN?cxt=filmography|access-date=2020-10-10|website=catalog.afi.com}}&lt;/ref&gt; The film debuted at No. 1 on the box office.&lt;ref&gt;{{cite news|last=Fox|first=David J.|date=October 12, 1993|title=Weekend Box Office Stallone, Snipes: Action at Box Office|work=[[Los Angeles Times]]|url=https://www.latimes.com/archives/la-xpm-1993-10-12-ca-44901-story.html|access-date=August 1, 2020}}&lt;/ref&gt;&lt;ref&gt;{{cite news|last=Fox|first=David J.|date=October 19, 1993|title=Weekend Box Office : 'Demolition Man' Fends Off 'Hillbillies'|work=[[Los Angeles Times]]|url=https://www.latimes.com/archives/la-xpm-1993-10-19-ca-47287-story.html|access-date=August 1, 2020}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|last=Horn|first=John|date=October 15, 1993|title=Demolition man' explodes into charts at no. 1|work=[[Chicago Tribune]]|url=https://www.chicagotribune.com/news/ct-xpm-1993-10-22-9310220292-story.html|access-date=May 4, 2020}}&lt;/ref&gt; and grossed $58,055,768 by the end of its box office run in North America and $159,055,768 worldwide.&lt;ref name=&quot;the-numbers&quot;&gt;{{cite web|title=Demolition Man - Box Office Data, Movie News, Cast Information|url=https://www.the-numbers.com/movie/Demolition-Man#tab=summary|access-date=June 1, 2020|work=[[The Numbers (website)|The Numbers]]}}&lt;/ref&gt;

853 Hulk Hogan popped up in ''Rocky III'', and in ''No Holds Barred'' he achieves the rare feat of making [[Sylvester Stallone]] seem sophisticated.

854 The fact that nothing about Mr. Hogan really adds up no doubt helps account for his popularity. His sober speaking voice outside of the ring does not match his wild roars when doing battle. The amused gleam in his eye hints at a canny intelligence behind the sinew and sweat. And his exaggeratedly stagy bouts make only a token attempt to look real. More than Sylvester Stallone or [[Arnold Schwarzenegger]], Mr. Hogan behaves like a self-invented [[comic-book]] character sprung to life.

855 Chuck Taylor All Star shoes have been worn by actors in feature films that include [[Jerry Lewis]] in ''[[The Colgate Comedy Hour]]'' (early 1950s), [[Elvis Presley]] in ''[[Change of Habit]]'' (1969), [[Sylvester Stallone]] in ''[[Rocky (film)|Rocky]]'' (1976), several cast members of ''[[Animal House]]'' (1978), basketball players in ''[[Grease (film)|Grease]]'' (1978), [[Tom Hanks]] in ''[[Bachelor Party (1984 film)|Bachelor Party]]'' (1984), [[Michael J. Fox]] in ''[[Back to the Future]]'' (1985),&lt;ref&gt;{{cite web |url=http://www.chucksconnection.com/back.html |title=Back to the Future |work=The ChucksConnection |access-date=2018-08-18 }}&lt;/ref&gt; the cast members of ''[[Hoosiers (film)|Hoosiers]]'' (1986), and [[Mike Myers]] and [[Dana Carvey]] in ''[[Wayne's World (film)|Wayne's World]]'' (1992). The video dust jacket of ''Hoosiers'' also featured the black high-tops.&lt;ref name=FamousFootwear&gt;{{cite news| first=Hal |last=Peterson | title =Famous Footwear | newspaper =[[The Republic (Columbus, Indiana)|The Republic]] | location =Columbus, Indiana | page=A-10 | date =March 7, 2001 }}&lt;/ref&gt;

856 Fifita is married to Dorothy Koloamatangi. They have a daughter, Vika; a son, [[Tanga Loa|Tevita]], who is also a wrestler; and two adopted sons, [[Tama Tonga|Alipate]] and [[Leo Tonga|Taula]]. His son Tevita played [[American football|football]] as a [[defensive end]] for the [[University of Texas at El Paso]] and was on the [[WWE]] roster from 2009 until 2014, as Camacho, and was also known as Micah in [[Total Nonstop Action Wrestling]]. Tevita currently wrestles as Tonga Loa and Alipate as Tama Tonga, both in WWE. Fifita's youngest son Taula&lt;ref&gt;{{cite web|url=http://www.orlandosentinel.com/sports/wrestling/os-wrestling-guerrillas-of-destiny-20170421-story.html|title=Guerrillas of Destiny, big in Japan, look for even more success|work=[[Orlando Sentinel]]|access-date=8 September 2017}}&lt;/ref&gt; made his professional wrestling debut in November 2016. He wrestles under the ring name Hikuleo&lt;ref&gt;{{cite web|url=http://www.f4wonline.com/japan/update-availability-new-japan-new-zealand-ippv-224491|title=Update on availability of New Japan New Zealand iPPV|work=[[Wrestling Observer Newsletter]]|access-date=3 December 2016|date=12 November 2016}}&lt;/ref&gt; previously, Leo Tonga.&lt;ref&gt;{{cite web|last1=Rose|first1=Bryan|title=Kenny Omega injures knee, still set for Kobe Destruction show|url=http://www.f4wonline.com/japan/kenny-omega-injures-knee-still-set-kobe-destruction-show-242331|website=Wrestling Observer Figure Four Online|access-date=6 September 2017|date=6 September 2017}}&lt;/ref&gt; Among Fifita's cousins are former [[Miami Dolphins]] defensive tackle [[Steve Fifita]] and [[Australia national rugby union team|Australian Wallabies]] rugby international player [[Tatafu Polota-Nau]]. Meng made a cameo appearance in the 1978 [[Sylvester Stallone]] movie ''[[Paradise Alley]]'' along with many other professional wrestlers.&lt;ref&gt;{{cite web|title=Full cast and crew for Paradise Alley (1978)|url=https://www.imdb.com/title/tt0078056/fullcredits#cast|website=IMDb|access-date=11 December 2012}}&lt;/ref&gt;

857 On the morning of Friday, May 13, 1949, a [[hazardous materials]] truck caught fire while passing through the [[Holland Tunnel]], which travels under the [[Hudson River]] between [[New York City]] and [[New Jersey]]. One firefighter was killed and 66 civilians were injured as a result of the fire. The 1996 motion picture [[Daylight (1996 film)|''Daylight'']], starring [[Sylvester Stallone]], was loosely based on this incident.

858 As early as the 1980s, Hugo Boss began with product placements and the outfitting of [[Celebrity|celebrities]]. Among other things, Hugo Boss outfitted the actors of the popular [[United States|US]] law series ''[[L.A. Law]]'' and was henceforth seen as the outfitter of [[yuppie]]s.&lt;ref&gt;{{Cite news |last=Clark |first=Andrew |date=2001-02-24 |title=Dressed for success |language=en-GB |work=[[The Guardian]] |url=https://www.theguardian.com/business/2001/feb/24/2 |access-date=2023-11-27 |issn=0261-3077}}&lt;/ref&gt; Hugo Boss dressed the leading actors [[Don Johnson]] and [[Philip Michael Thomas]] in the crime series [[Miami Vice]].&lt;ref name=&quot;:17&quot; /&gt; Other well-known personalities wore ''Boss'' outfits at the time, such as [[Michael Jackson]], who wore a white ''Boss'' suit on the album cover of [[Thriller (album)|Thriller]],&lt;ref&gt;{{Cite news |last=Sack |first=Adriano |date=2018-09-02 |title=Der weiße Anzug |work=[[Welt am Sonntag]] |pages=61 | language=de |volume=35}}&lt;/ref&gt; or [[Sylvester Stallone]], who wore a ''Boss'' sweater as [[Rocky Balboa|Rocky]].&lt;ref&gt;{{Cite news |last=Meier |first=Ernst |date=2015-06-10 |title=Die Stadt Zug steht Hugo Boss gut. |work=Neue Zuger Zeitung |pages=45 |language=de}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last1=Emig |first1=Silke |last2=Bayer |first2=Tobias |date=2021-09-20 |title=„Da passen wir wahnsinnig auf“ |work=Textilwirtschaft |pages=50-51 |language=de}}&lt;/ref&gt;

859 San Angeles was first conceived as a setting for the 1982 movie ''[[Blade Runner]]'' in an early script.&lt;ref&gt;Turan, Kenneth. (September 13, 1992) [[Los Angeles Times]] ''Blade Runner 2. The screenwriter wrote eight drafts -- and then was replaced.'' Section: TM-Los Angeles Times Magazine; Page 19.&lt;/ref&gt; Ten years later, the 1993 script for the 1994 movie ''[[Double Dragon (film)|Double Dragon]]'', post-earthquake California merged Los Angeles and San Diego into one [[Megalopolis (city type)|megalopolis]] called San Angeles, half of which was under water.&lt;ref&gt;Honeycutt, Kirk. (March 3, 1993) [[The Hollywood Reporter]] ''&quot;Double Dragon&quot; enters film game.'' Page 3.&lt;/ref&gt; At about the same time, the San Angeles concept city also was used in the 1993 movie ''[[Demolition Man (film)|Demolition Man]]'',&lt;ref&gt;Elliott, Stuart. (October 8, 1993) [[New York Times]] ''[https://query.nytimes.com/gst/fullpage.html?res=9F0CEEDF153CF93BA35753C1A965958260[The Media Business: Advertising; In Demolition Man, a car could be your grandson's Oldsmobile.]'' Section D.15.&lt;/ref&gt; where the earthquake-destroyed Los Angeles of 2010 was replaced by the city San Angeles that stretched from San Diego to Santa Barbara.&lt;ref name=&quot;snipes&quot;&gt;Wong, Cy. (April 16, 1993) [[Los Angeles Times]] ''Movie: Action-thriller starring Wesley Snipes and Sylvester Stallone is being filmed in the city this week.'' Section: ME-Metro; Page 1.&lt;/ref&gt;&lt;ref name=&quot;Wong&quot;&gt;{{cite news |title=Irvine Cast as Futuristic L.A. : Movie: Action-thriller starring Wesley Snipes and Sylvester Stallone is being filmed in the city this week.|work=[[Los Angeles Times]] |date=April 16, 1993 |url=https://www.latimes.com/archives/la-xpm-1993-04-16-me-23636-story.html |access-date=November 8, 2010 |first=Stacy |last=Wong }}&lt;/ref&gt;&lt;ref&gt;Thomas, Bob. (October 7, 1993) [[Associated Press]]. ''AP Arts: Film Review-Demolition Man.'' Section: Entertainment.&lt;/ref&gt; The Demolition Man/San Angeles, a modified [[Irvine, California]], set in the year 2032,&lt;ref&gt;Arnold, Gary. (October 7, 1993) [[Washington Times]] ''Washington weekend movies; Movie mini-reviews.'' Section: M24&lt;/ref&gt; maintained a police force called &quot;SaPD&quot; (for San Angeles Police Department), which used black and white, gull-winged cars having SaPD emblazoned on them.&lt;ref name=&quot;snipes&quot;/&gt; The 1993 ''Demolition Man'' movie also set the fictional Arnold

Schwarzenegger Presidential Museum in San Angeles,&lt;ref&gt;Harvey, Steve. (May 17, 2002) [[Los Angeles Times]] ''Only in L.A.'' Section: California Metro; page 4.&lt;/ref&gt; even though [[Arnold Schwarzenegger¦Arnold Schwarzenegger's]] [[Political career of Arnold Schwarzenegger#First years: 2003-2004¦first run for political office]] would not be for another ten years.

860 [[Category:Cultural depictions of Sylvester Stallone]]

861 Jones has appeared in a number of television and movie roles. In May 2009, Jones was cast in the star-studded feature film [[The Expendables (2010 film)¦The Expendables]] opposite Sylvester Stallone and Mickey Rourke as the character, Cheyenne, Mickey's love interest in the film. The film was released in theaters August 13, 2010. Jones guest appeared on ''[[Rescue Me (U.S. TV series)¦Rescue Me]]'' in July 2008 and on the [[Nickelodeon]] TV series ''[[The Naked Brothers Band (TV series)¦The Naked Brothers Band]]'' as Double X. Jones has also appeared on [[MTV]]2's ''[[Wonder Showzen]]'', the [[soap opera]] ''[[Guiding Light]]'', in the feature film ''[[Spider-Man 3]]'', and as a [[Barker's Beauties¦Barker's Beauty]] on the [[CBS]]' [[game show]] ''[[The Price Is Right (U.S. game show)¦The Price Is Right]]'', where she appeared as a model for five weeks in April and May 2007. In late 2004 - early 2005, Jones participated in skits during the first ever SmackDown! Rookie Diva of the Year contest at [[No Way Out (2005)¦No Way Out 2005]]. Jones starred in Fox's short lived series ''Anchorwoman''. The show debuted August 22, 2007 on FOX.&lt;ref&gt;{{cite web¦last1=Malone¦first1=Michael¦last2=Grossman¦first2=Ben¦title=Fox Cancels Anchorwoman¦url=http://www.broadcastingcable.com/article/CA6471406.html¦website=broadcastingcable.com¦accessdate=19 June 2017¦archiveurl=https://web.archive.org/web/20080708223728/http://www.broadcastingcable.com/article/CA6471406.html¦archivedate=July 8, 2008¦date=August 23, 2007}}&lt;/ref&gt;

862 * Every square inch of her refrigerator door at her farm house is covered with a personal photo gallery of famous people she has met over the years, including [[Molly Ringwald]], [[Sylvester Stallone]], [[Meat Loaf]], Bruce Springsteen, [[Jon Lovitz]], Stephen Huff and [[Sean Penn]].&lt;ref name=Paste/&gt;{{citation needed¦date=February 2012}}

863 Roth played [[Germany national football team¦Germany national team]] [[Captain (association football)¦captain]] Baumann in the [[1981 in film¦1981 movie]] ''[[Escape to Victory¦Victory]]'' (titled ''Escape to Victory'' in Europe), which also featured former Cosmos teammate [[Pelé]], as well as [[Sylvester Stallone]] and [[Michael Caine]]. He was inducted into the [[National Soccer Hall of Fame]] in 1989.&lt;ref&gt;{{Cite web ¦title=Werner Roth - 1989 Inductee {{!}} National Soccer Hall of Fame ¦url=http://www.nationalsoccerhof.com/players/werner-roth.html ¦access-date=2023-12-19 ¦website=Werner Roth - 1989 Inductee {{!}} National Soccer Hall of Fame ¦language=en}}&lt;/ref&gt;

864 ¦ [[Sylvester Stallone]]: What's next for the star of [[Rocky]]?

865 ¦ [[Sylvester Stallone]]: A Chat with the [[Rocky III]] Champ!

866 ** [[Creed (film)¦''Creed'' (film)]], a 2015 spin-off/sequel to the ''Rocky'' film series, starring Michael B. Jordan and Sylvester Stallone

867 ** ''[[Creed II]]'', a sequel to ''Creed'', also starring Michael B. Jordan and Sylvester Stallone

868 * In the 1996 film ''[[Daylight (1996 film)¦Daylight]]'', directed by [[Rob Cohen]] and starring [[Sylvester Stallone]], there are references to &quot;sandhogs&quot; as well as detailed history on how these men lived under the pressures of building the Holland Tunnel.[https://www.imdb.com/title/tt0116040/]

869 Two-liter bottles and six-packs of cans of Pepsi Free appear in a refrigerator case behind Sylvester Stallone's character in the grocery store scene in the 1986 film ''[[Cobra (1986 film)¦Cobra]]''.

870 There was a brief period when the episode had a different subplot that revolved around the restaurant chain [[Planet Hollywood]]. Groening had been told by a spokesperson that if he put Planet Hollywood in ''The Simpsons'', the creators of the restaurant, [[Arnold Schwarzenegger]], [[Bruce Willis]] and [[Sylvester Stallone]], would agree to make guest appearances on the show. The writers of ''The Simpsons'' were excited about this so they wrote a new subplot for the episode that featured Planet Hollywood and the three actors. However, for unknown reasons, they were unable to appear in the episode.&lt;ref name=&quot;Groening&quot;/&gt;

871 *[[Sylvester Stallone]]

872 * [[Sylvester Stallone]]

873 '''''Escape to Victory''''' (or simply '''''Victory''''') is a 1981 [[Sports film¦sports]] [[war film]]&lt;ref name=&quot;USUKIT&quot;&gt;{{Cite web¦url=https://filmarchiv.hu/hu/filmmaraton/2019/filmek/menekules-a-gyozelembe¦title=Menekülés a győzelembe - Budapesti Klasszikus Film Maraton¦website=Nemzeti Filmintézet - Filmarchívum¦access-date=March 9, 2021¦archive-date=October 30, 2020¦archive-url=https://web.archive.org/web/20201030232922/https://filmarchiv.hu/hu/filmmaraton/2019/filmek/menekules-a-gyozelembe¦url-status=live}}&lt;/ref&gt; directed by [[John Huston]] and starring [[Sylvester Stallone]], [[Michael Caine]], [[Max von Sydow]] and [[Pelé]]. The film is about [[Allies of World War II¦Allied]] [[prisoners of war]] who are interned in a [[Nazi Germany¦German]] prison camp during the [[World War II¦Second World War]] who play an exhibition match of [[Association football¦football]] against a German team.

874 The film received great attention upon its theatrical release, as it starred professional [[Association football¦footballers]] [[Bobby Moore]], [[Osvaldo Ardiles]], [[Kazimierz Deyna]], [[Paul Van Himst]], [[Mike Summerbee]], [[Hallvar Thoresen]], [[Werner Roth (soccer, born 1948)¦Werner Roth]] and Pelé. Numerous [[Ipswich Town F.C.¦Ipswich Town]] players were also in the film, including [[John Wark]], [[Russell Osman]], [[Laurie Sivell]], [[Robin Turner (footballer)¦Robin Turner]] and [[Kevin O'Callaghan]]. Other Ipswich Town players stood in for actors in the football scenes &amp;ndash; [[Kevin Beattie]] for Michael Caine, and [[Paul Cooper (footballer, born 1953)¦Paul Cooper]] for Sylvester Stallone. Yabo Yablonsky wrote the script and the film was entered into the [[12th Moscow International Film Festival]].&lt;ref name=&quot;Moscow1981&quot;&gt;{{cite web ¦url=http://www.moscowfilmfestival.ru/miff34/eng/archives/?year=1981 ¦title=12th Moscow International Film Festival (1981) ¦access-date=January 27, 2013 ¦website=MIFF ¦url-status=dead ¦archive-url=https://web.archive.org/web/20130421050907/http://www.moscowfilmfestival.ru/miff34/eng/archives/?year=1981 ¦archive-date=April 21, 2013}}&lt;/ref&gt;

875 * [[Sylvester Stallone]] as Captain Robert Hatch (U.S.A.)

876 ''Escape to Victory'' featured a great many professional footballers as both the POW team and the German team. Many of the footballers came from the [[Ipswich Town F.C.¦Ipswich Town]] squad, who were at the time one of the most [[UEFA Cup¦successful]] teams in Europe.&lt;ref&gt;{{Cite news¦url=https://www.bbc.com/news/uk-england-suffolk-57840345¦title=Escape to Victory: The Ipswich footballers who made a cult classic¦work=BBC News¦date=July 24, 2021¦access-date=July 25, 2021¦archive-date=July 25, 2021¦archive-url=https://web.archive.org/web/20210725052402/https://www.bbc.com/news/uk-england-suffolk-57840345¦url-status=live}}&lt;/ref&gt; Despite not appearing on screen, English World Cup-winning goalkeeper [[Gordon Banks]] and Alan Thatcher were closely involved in the film, working with [[Sylvester Stallone]] on his goalkeeping scenes. ''[[Sports Illustrated]]'' magazine said &quot;the game is marvelously photographed by [[Gerry Fisher]], under second unit director [[Robert Riger]].&quot;&lt;ref&gt;{{cite magazine¦ url=https://www.si.com/vault/1981/08/10/825842/pow-right-in-the-kisser-seeking-soccer-quotvictoryquot-over-pows-the-germans-stop-playing-mr-nice-guy¦ magazine=[[Sports Illustrated]]¦ title=P.O.W., Right In The Kisser¦ last=Deford¦ first=Frank¦ date=August 10, 1981¦ access-date=August 19, 2018¦ archive-date=June 5, 2022¦ archive-url=https://web.archive.org/web/20220605091118/https://vault.si.com/vault/1981/08/10/pow-right-in-the-kisser-seeking-soccer-quotvictoryquot-over-pows-the-germans-stop-playing-mr-nice-guy¦ url-status=live}}&lt;/ref&gt;

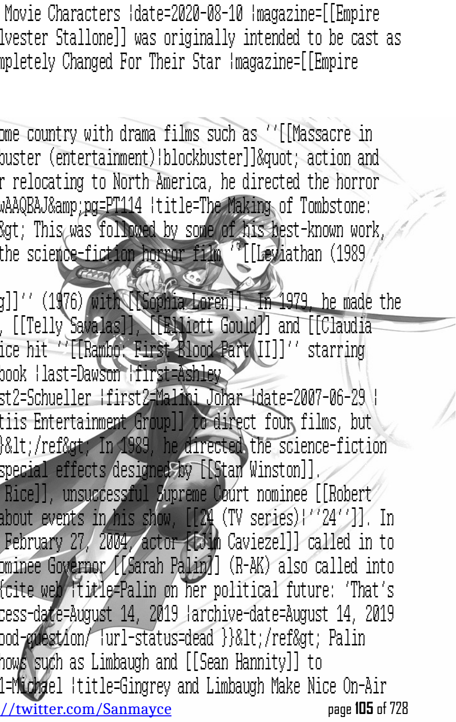
877 Detective '''Axel James''' '''Foley''' is a fictional character, portrayed by [[Eddie Murphy]], and is the titular [[protagonist]] of the ''[[Beverly Hills Cop (franchise)¦Beverly Hills Cop]]'' film series.&lt;ref&gt;{{cite news¦title= Eddie Murphy: Hollywood's $2 billion man talks about his marriage, 'Beverly Hills Cop III' and the joys of fatherhood ¦work=[[Ebony (magazine)¦Ebony]]¦ first=Walter ¦last=Leavy¦date=June 1994¦url= https://books.google.com/books?id=6GntfRruQ7IC&amp;q=Eddie+Murphy%3A+Hollywood%27s+%242+billion+man+talks+about+his+marriage%2C+%27Beverly+Hills+Cop+III%27+and+the+joys+of+fatherhood&amp;pg=PA100¦access-date=November 13, 2010 ¦pages=100-106}}&lt;/ref&gt;&lt;ref&gt;{{cite web ¦url=http://www.citypages.com/music/7-twin-cities-hip-hop-albums-you-might-have-missed-7487681 ¦url-status=dead ¦archive-url=https://web.archive.org/web/20150721001412/http://www.citypages.com/music/7-twin-cities-hip-hop-albums-you-might-have-missed-7487681 ¦archive-date=2015-07-21 ¦title=7 Twin Cities hip-hop albums you might have missed {{!}} City Pages}}&lt;/ref&gt; He is ranked {{Abbr¦No.¦Number}} 55 on ''[[Empire (film magazine)¦Empire]]'' magazine's 2008&lt;ref&gt;{{cite magazine ¦url=https://www.empireonline.com/100-greatest-movie-characters/default.asp?c=78 ¦title=The 100 Greatest Movie Characters ¦date=2008-12-01 ¦magazine=[[Empire (magazine)¦Empire]] ¦access-date=May 21, 2010 ¦archive-url=https://web.archive.org/web/20110905194537/https://www.empireonline.com/100-greatest-movie-characters/default.asp?c=78 ¦archive-date=2011-09-05 ¦url-status=dead}}&lt;/ref&gt; and 2020 lists of ''The 100 Greatest Movie Characters''.&lt;ref&gt;{{cite magazine ¦title=The 100 Greatest Movie Characters ¦date=2020-08-10 ¦magazine=[[Empire (magazine)¦Empire]] ¦url=https://www.empireonline.com/movies/features/100-greatest-movie-characters/ ¦access-date=2024-07-04}}&lt;/ref&gt; [[Sylvester Stallone]] was originally intended to be cast as Axel Foley.&lt;ref&gt;{{cite magazine ¦url= https://www.empireonline.com/features/recasting-blockbusters/ ¦title= Re-Cast: Five Blockbusters Completely Changed For Their Star ¦magazine=[[Empire (magazine)¦Empire]] ¦access-date=November 13, 2010}}&lt;/ref&gt;

878 File:Flickr - nicogenin - 66ème Festival de Venise (Mostra) - Sylvester Stallone (26).jpg¦[[Sylvester Stallone]] American Actor

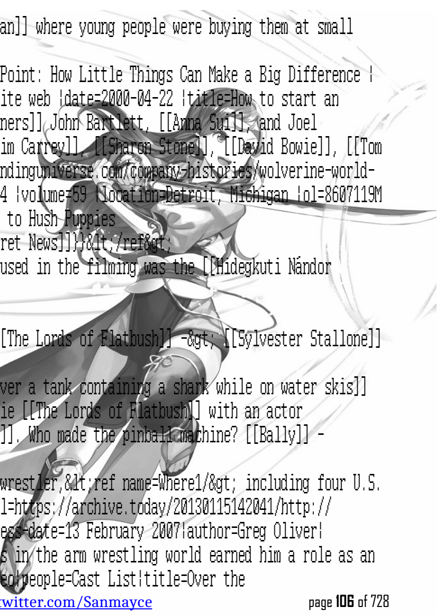
879 '''George Pan Cosmatos''' (4 January 1941 - 19 April 2005) was a Greek-Italian film director and screenwriter. Following early success in his home country with drama films such as ''[[Massacre in Rome]]'' with [[Richard Burton]] (based on the real-life [[Ardeatine massacre]]), Cosmatos retooled his career towards mainstream &quot;[[Blockbuster (entertainment)¦blockbuster]]&quot; action and adventure films, including ''[[The Cassandra Crossing]]'' and ''[[Escape to Athena]]'', both of which were British-Italian co-productions. After relocating to North America, he directed the horror film ''[[Of Unknown Origin]]''.&lt;ref name=&quot;:0&quot;&gt;{{Cite book ¦last=Farkis ¦first=John ¦url=https://books.google.com/books?id=kH56DwAAQBAJ&amp;pg=PT114 ¦title=The Making of Tombstone: Behind the Scenes of the Classic Modern Western ¦date=2018-11-26 ¦publisher=McFarland ¦isbn=978-1-4766-7586-2 ¦pages=106 ¦language=en}}&lt;/ref&gt; This was followed by some of his best-known work, including the action films ''[[Rambo: First Blood Part II]]'' and ''[[Cobra (1986 film)¦Cobra]]'' (both of which star [[Sylvester Stallone]]), the science-fiction horror film ''[[Leviathan (1989 film)¦Leviathan]]'', and the critically acclaimed [[Western film¦Western movie]] [[Tombstone (film)¦''Tombstone'']].

880 Cosmatos found success in Italy for directing the films ''[[Rappresaglia]]'' (1973) with [[Marcello Mastroianni]] and ''[[The Cassandra Crossing]]'' (1976) with [[Sophia Loren]]. In 1979, he made the British [[World War II]] adventure film ''[[Escape to Athena]]'', starring an all-star ensemble cast including [[Roger Moore]], [[David Niven]], [[Telly Savalas]], [[Elliott Gould]] and [[Claudia Cardinale]]. He made his North American directorial debut with the Canadian horror film ''[[Of Unknown Origin]]''. He then directed the box-office hit ''[[Rambo: First Blood Part II]]'' starring [[Sylvester Stallone]], and ''[[Cobra (1986 film)¦Cobra]]'', another Stallone film, in 1986.&lt;ref name=&quot;:0&quot; /&gt;&lt;ref&gt;{{Cite book ¦last=Dawson ¦first=Ashley ¦url=https://books.google.com/books?id=sKTiyHCT9KQC&amp;pg=PA46 ¦title=Exceptional State: Contemporary U.S. Culture and the New Imperialism ¦last2=Schueller ¦first2=Malini Johar ¦date=2007-06-29 ¦publisher=Duke University Press ¦isbn=978-0-8223-8964-4 ¦pages=46 ¦language=en}}&lt;/ref&gt; On 8 August 1986, Cosmatos agreed with [[De Laurentiis Entertainment Group]] to direct four films, but none would be made.&lt;ref&gt;{{Cite news¦date=1986-08-06¦title=DEG Inks Cosmatos To Five-Year Pact¦page=5¦work=[[Variety (magazine)¦Variety]]}}&lt;/ref&gt; In 1989, he directed the science-fiction horror film [[Leviathan (1989 film)¦''Leviathan'']], starring [[Peter Weller]], [[Richard Crenna]], [[Ernie Hudson]] and [[Amanda Pays]], with special effects designed by [[Stan Winston]].

881 Other notable guests who called in to Limbaugh's show include former Secretary of Defense [[Donald Rumsfeld]], Secretary of State [[Condoleezza Rice]], unsuccessful Supreme Court nominee [[Robert Bork]], economist [[Thomas Sowell]], former Federal Reserve Chairman [[Alan Greenspan]], and television writer [[Joel Surnow]], who took calls about events in his show, [[24 (TV series)¦''24'']]. In December 2006, [[Sylvester Stallone]] made an appearance on the show to discuss his upcoming movie ''[[Rocky Balboa (film)¦Rocky Balboa]]''. On February 27, 2004, actor [[Jim Caviezel]] called in to the program to discuss ''[[The Passion of the Christ]]'', in which Caviezel played the role of [[Jesus]] Christ. Republican vice presidential nominee Governor [[Sarah Palin]] (R-AK) also called into a show before a rally in October 2008 to discuss the election and the economic distortion and impact of Senator Obama's tax policy;&lt;ref&gt;{{cite web ¦title=Palin on her political future: 'That's a good question' ¦url=http://politicalticker.blogs.cnn.com/2008/10/14/palin-on-her-political-future-thats-a-good-question/ ¦website=cnn.com ¦access-date=August 14, 2019 ¦archive-date=August 14, 2019 ¦archive-url=https://web.archive.org/web/20190814184447/http://politicalticker.blogs.cnn.com/2008/10/14/palin-on-her-political-future-thats-a-good-question/ ¦url-status=dead }}&lt;/ref&gt; Palin returned to the show in November 2009 to discuss her book ''[[Going Rogue: An American Life]]''. [[Phil Gingrey]], a congressman who compared shows such as Limbaugh and [[Sean Hannity]] to &quot;throwing bricks&quot; in January 2009, gave an interview on Limbaugh's show the following day.&lt;ref&gt;{{cite web ¦last1=O'Brien ¦first1=Michael ¦title=Gingrey and Limbaugh Make Nice On-Air



```
    Weds ¦url=https://thehill.com/blogs/blog-briefing-room/news/39282-gingrey-and-limbaugh-make-nice-on-air-weds ¦website=thehill.com ¦date=January 28, 2009 ¦publisher=Capitol Hill Publishing Corp. ¦access-date=February 22, 2019 ¦archive-date=February 23, 2019 ¦archive-url=https://web.archive.org/web/20190223074608/https://thehill.com/blogs/blog-briefing-room/news/39282-gingrey-and-limbaugh-make-nice-on-air-weds ¦url-status=live }}&lt;/ref&gt;
882 ¦ creator = [[Sylvester Stallone]]
883 '''James''' &quot;'''Clubber'''&quot; '''Lang''' is a fictional character created by [[Sylvester Stallone]] for the film ''[[Rocky III]]'', which was released in 1982.&lt;ref&gt;{{cite book¦url=https://books.google.com/books?id=OBQNk53ai9MC&amp;pg=PA94 ¦title=Movies and the Reagan Presidency: Success and Ethics ¦author=Chris Jordan ¦year=2003 ¦page=94 ¦publisher=Bloomsbury Academic ¦isbn=9780275979676 }}&lt;/ref&gt; He was portrayed by American actor and real-life wrestler [[Mr. T]] and serves as the main [[antagonist]] of the film.&lt;ref&gt;{{cite book¦url=https://books.google.com/books?id=27KIBUeUIPQC&amp;pg=PA285 ¦title=Western Fictions, Black Realities: Meanings of Blackness and Modernities ¦author1=Isabel Soto ¦author2=Violet Showers Johnson ¦year=2011 ¦page=285 ¦publisher=LIT Verlag Münster ¦isbn=9783643901095 }}&lt;/ref&gt;
884 ¦ voices             = Reo Jones&lt;br /&gt;Chris Lundquist&lt;br /&gt;[[Kathleen Barr]]&lt;br /&gt;D. Kevin Williams&lt;br /&gt;[[Walter Cronkite]]&lt;br /&gt;[[Sylvester Stallone]]&lt;br /&gt;[[Ben Stiller]]&lt;br /&gt;[[Billy Crystal]]&lt;br /&gt;[[Annette Bening]]&lt;br /&gt;[[Dustin Hoffman]]&lt;br /&gt;[[Michael Douglas]]&lt;br /&gt;[[Arnold Schwarzenegger]]&lt;br /&gt;[[Liam Neeson]]&lt;br /&gt;[[Whoopi Goldberg]]
885 The show features celebrity voice talents, such as [[CBS News]] anchorman [[Walter Cronkite]] (as [[Benjamin Franklin]]), [[Sylvester Stallone]] (as [[Paul Revere]]), [[Ben Stiller]] (as [[Thomas Jefferson]]), [[Billy Crystal]] (as [[John Adams]]), [[Annette Bening]] (as [[Abigail Adams]]), [[Dustin Hoffman]] (as [[Benedict Arnold]]), [[Michael Douglas]] (as [[Patrick Henry]]), [[Arnold Schwarzenegger]] (as [[Friedrich Wilhelm von Steuben¦Baron von Steuben]]), [[Liam Neeson]] (as [[John Paul Jones]]), [[Whoopi Goldberg]] (as [[Deborah Sampson]]), [[Charles Shaughnessy]] (as King [[George III]]), [[Michael York]] (as Admiral Lord [[Richard Howe, 1st Earl Howe¦Richard Howe]]), [[Ralph Fiennes]] (as General Lord [[Charles Cornwallis, 1st Marquess Cornwallis¦Charles Cornwallis]]), [[Don Francisco (television host)¦Don Francisco]] (as [[Bernardo de Gálvez]]), and [[Aaron Carter]] (as [[Joseph Plum Martin]]) who lend credence to characters critical to the forming of a free country, from the [[Boston Tea Party]] to the [[Constitutional Convention (United States)¦Constitutional Convention]].
886 * [[Paul Revere]] ([[Sylvester Stallone]])
887 Muscio connects the prominence of Italian-American [[director-actor]]s to the ''{{ill¦Capocomico¦it}}'' of traditional Italian theatre. Such director-actors include: [[Danny DeVito]], [[John Turturro]], [[Steve Buscemi]], [[Vincent Gallo]], [[Al Pacino]], [[Stanley Tucci]], [[Robert De Niro]], [[Sylvester Stallone]], [[Gary Sinise]], [[Anne Bancroft]], and [[Madonna (entertainer)¦Madonna]].&lt;ref name=&quot;Muscio&quot; /&gt;
888 *[[Sylvester Stallone]] (born 1946) film actor, director, producer, and screenwriter, best known for [[Rocky (franchise)¦Rocky]] and [[Rambo (franchise)¦Rambo]]&lt;ref&gt;https://movies.yahoo.com/movie/contributor/1800020291 &quot;stoic Italian-American actor&quot;&lt;/ref&gt;
889 * [[Sylvester Stallone]], actor (''[[Rocky (film series)¦Rocky]]'', ''[[Rambo (film series)¦Rambo]]'', ''[[The Expendables (film series)¦The Expendables]]'' franchises)&lt;ref name=Mummers/&gt;
890 * Various members of the original Guardians of the Galaxy team appear in the live-action film ''[[Guardians of the Galaxy Vol. 2]]'', with [[Michael Rooker]] as [[Yondu¦Yondu Udonta]], [[Sylvester Stallone]] as [[Starhawk (character)¦Stakar Ogord]], [[Michael Rosenbaum]] as [[Martinex¦Martinex T'Naga]], [[Ving Rhames]] as [[Charlie-27]], [[Michelle Yeoh]] as [[Aleta Ogord]], an uncredited [[Miley Cyrus]] as the voice of [[Mainframe (comics)¦Mainframe]], and Krugarr in a non-speaking role. These versions of the characters are part of an interstellar group of thieves, smugglers, and pirates known as the [[Ravagers (Marvel Cinematic Universe)¦Ravagers]].&lt;ref name=&quot;OldTeam&quot;&gt;{{cite web¦url=http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu¦title=Sylvester Stallone's 'Guardians of the Galaxy Vol. 2' character has a big future in the MCU¦last=Daniell¦first=Mark¦work=The Whig¦date=April 25, 2017¦access-date=April 28, 2017¦archive-url=https://web.archive.org/web/20170428051803/http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu¦archive-date=April 28, 2017¦url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;OldTeam2&quot;&gt;{{cite web¦url=http://screenrant.com/guardians-of-the-galaxy-2-original-team/¦title=Kevin Feige &amp; James Gunn Explain Stallone's Secret MCU Team¦last=Keyes¦first=Rob¦publisher=Screen Rant¦date=April 20, 2017¦access-date=April 28, 2017¦archive-url=https://web.archive.org/web/20170422033433/http://screenrant.com/guardians-of-the-galaxy-2-original-team/¦archive-date=April 22, 2017¦url-status=live}}&lt;/ref&gt;
891 {{Infobox¦decat=yes¦child=yes¦label1=Additional dialogue by¦data1={{Ubl¦[[Sylvester Stallone]]}}}}
892 * Sylvester Stallone
893 '''''The Lords of Flatbush''''' (stylized on-screen as '''''The Lord's of Flatbush''''') is a 1974 American [[comedy]] directed by [[Martin Davidson]] and [[Stephen Verona¦Stephen F. Verona]]. The film stars [[Sylvester Stallone]], [[Perry King]], Paul Mace, [[Henry Winkler]], and [[Susan Blakely]]. Stallone was also credited with writing additional dialogue. The plot is about street teenagers in [[Greaser (subculture)¦leather jacket]]s from the [[Flatbush, Brooklyn¦Flatbush]] neighborhood of [[Brooklyn, New York]].&lt;ref&gt;{{cite web ¦url=http://thenewbev.com/blog/2019/01/the-lords-of-flatbush/ ¦title=The Lords of Flatbush ¦first=Quentin ¦last=Tarantino ¦author-link=Quentin Tarantino ¦publisher=[[New Beverly Cinema]] ¦date=2019-01-07 ¦access-date=2020-03-11}}&lt;/ref&gt; The movie, along with ''[[American Graffiti]]'', the television hit ''[[Happy Days]]'', the musical ''[[Grease (musical)¦Grease]]'' and its [[Grease (film)¦like-named film]] version, and novelty rock act "[[Sha Na Na]]'', was part of a resurgence in popular interest in the '50s greaser culture in the 1970s.&lt;ref&gt;{{Cite web ¦last=Prigge ¦first=Matthew J. Prigge ¦date=December 28, 2016 ¦title=Greaser Redux: The 1970s Revival of the &quot;Greaser ¦url=https://shepherdexpress.com/culture/milwaukee-history/greaser-redux-1970s-revival-greaser-x93-milwaukee/ ¦website=www.shepherdexpress.com}}&lt;/ref&gt;
894 * [[Sylvester Stallone]] as Stanley Rosiello
895 {{Sylvester Stallone}}
896 Artists that Murphy has recorded and/or produced for include [[Dolly Parton]], [[Sylvester Stallone]], [[Jennifer Warnes]], [[Siedah Garrett]], [[Timothy B. Schmit¦Tim Schmitt]], [[Richie Havens]], [[Isaac Hayes]], [[Taj Mahal (musician)¦Taj Mahal]], [[Paul Williams (songwriter)¦Paul Williams]], and [[Kenny Rankin]]. He was also a guest conductor for the [[Dallas Symphony Orchestra]], during their successful Summer Pops series.&lt;ref name=&quot;Airedel&quot;&gt;{{Cite web¦url=http://www.air-edel.co.uk/usroster/wmurphy.htm¦title=wmurphy¦date=January 16, 2003¦archive-url=https://web.archive.org/web/20030116011814/http://www.air-edel.co.uk/usroster/wmurphy.htm¦access-date=September 30, 2021¦archive-date=January 16, 2003}}&lt;/ref&gt;
897 In 1993, per producer Joel Silver, Cortese made two separate cameos ([[Taco Bell]] Lounge singer and a Cryo Prison guard) in the hit [[Sylvester Stallone]] film ''[[Demolition Man (film)¦Demolition Man]]''. Also, in 1993, Cortese landed the lead role in NBC's remake of &quot;Route 66&quot;. In 1994, he co-starred in the CBS police drama ''[[Traps (TV series)¦Traps]]'' with Academy Award winner [[George C. Scott]]. He gained further recognition in 1995 as Jess Hanson in ''[[Melrose Place]]''.&lt;ref&gt;{{cite web¦url=https://www.imdb.com/name/nm0004844/¦title=Dan Cortese¦publisher=IMDb.com¦access-date=2015-03-04}}&lt;/ref&gt;  He played Jake Hanson's half brother for 8 episodes in the third season but he was killed off in that season's cliffhanger finale.
898 Lange was a contestant on the first season of the NBC reality TV show, [[The Contender (TV series)¦The Contender]], produced and hosted by [[Sylvester Stallone]] and [[Sugar Ray Leonard]]. Stallone's brother attended one of Jimmy's boxing matches in Washington DC and was so impressed, he recommended him for the show as one of the 16 boxers.  Current special adviser, &quot;First Lady of Boxing&quot;  [[Jackie Kallen]] served as 'den mother' on the show.
899 In 2006, she appeared in a 30-minute film, ''[[Vic (film)¦Vic]]'', co-written and directed by [[Sage Stallone]], the son of [[Sylvester Stallone]].&lt;ref name=&quot;variety&quot;/&gt;&lt;ref name=&quot;Movieweb&quot;&gt;{{Cite web ¦title=Carol Lynley, 'The Poseidon Adventure' &amp; 'Blue Denim' Star, Dies at 77¦url=https://movieweb.com/carol-lynley-dead-actress/¦website=Movieweb¦access-date=September 8, 2019¦date=September 6, 2019}}&lt;/ref&gt;
900 Santoni again collaborated with Reiner in the comedic homage to film noir ''[[Dead Men Don't Wear Plaid]]'' (1982).&lt;ref name=Alexander/&gt; His other film roles during this decade included juvenile correction facility officer Ramon Herrera in ''[[Bad Boys (1983 film)¦Bad Boys]]'' (1983), as well as Detective Tony Gonzales in the action film ''[[Cobra (1986 film)¦Cobra]]'' (1986) opposite [[Sylvester Stallone]]. He featured in the short-lived series ''[[Sanchez of Bel Air]]'' and ''[[Manimal]]'',&lt;ref name=Barnes/&gt; in which he played Nick Rivera.&lt;ref name=&quot;TVG Filmography&quot;/&gt; He was described by [[Tracy Newman]] as having a completely different personality from the characters he played, which were most frequently cops, crime lords, or judges. She noted his reputation among friends as being &quot;the funniest guy in the room&quot;.&lt;ref name=Alexander/&gt;
901 In 1994, when sales were down to 30,000 pairs a year, Hush Puppies suddenly became [[hip (slang)¦hip]] in the clubs and bars of downtown [[Manhattan]] where young people were buying them at small shoe stores.&lt;ref name=&quot;TippingPoint&quot;&gt;{{Cite book ¦last=Gladwell ¦first=Malcolm ¦url=https://books.google.com/books?id=yBDBEGBIUmgC&amp;q=Hush+Puppies+won+the+prize+for+best+accessory+at+the+Council+of+Fashion+Designers+awards+dinner+in+1996. ¦title=The Tipping Point: How Little Things Can Make a Big Difference ¦date=2000 ¦publisher=[[Little, Brown and Company]] ¦isbn=978-0-7595-7473-1 ¦author-link=Malcolm Gladwell ¦access-date=June 11, 2015}}&lt;br&gt;{{Cite web ¦date=2000-04-22 ¦title=How to start an epidemic ¦url=http://www.theguardian.com/books/2000/apr/22/extract ¦access-date=2022-11-02 ¦website=[[The Guardian]]}}&lt;/ref&gt; [[Fashion designers]] John Bartlett, [[Anna Sui]], and Joel Fitzpatrick began featuring them in their collections; the shoes were soon worn by celebrities such as [[Kenneth Clarke]], [[Princess Diana]], [[Jim Carrey]], [[Sharon Stone]], [[David Bowie]], [[Tom Hanks]], [[Ellen DeGeneres]] and [[Sylvester Stallone]].&lt;ref name=&quot;:0&quot;&gt;{{Cite book ¦last=Pederson ¦first=Jay P. ¦url=http://www.fundinguniverse.com/company-histories/wolverine-world-wide-inc-history/ ¦title=International Directory of Company Histories ¦date=February 6, 2004 ¦publisher=[[St. James Press]] ¦isbn=978-1-55862-504-4 ¦volume=59 ¦location=Detroit, Michigan ¦ol=8607119M ¦via=Funding Universe}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦last=Associated Press ¦date=1996-05-12 ¦title=Trendy Feet Find Themselves Beating a Path to Hush Puppies ¦url=https://www.deseret.com/1996/5/12/19242001/trendy-feet-find-themselves-beating-a-path-to-hush-puppies ¦access-date=2022-11-02 ¦website=[[Deseret News]]}}&lt;/ref&gt;
902 The stadium was portrayed in the 1981 film ''[[Escape to Victory]]'' starring [[Sylvester Stallone]] and [[Michael Caine]], but the stand-stadium used in the filming was the [[Hidegkuti Nándor Stadion (1947)]] in [[Budapest]], [[Hungary]].
903 [[Category:Images of Sylvester Stallone]]</text>
904       <text bytes="1325" sha1="fdfhjx79wtkm99995w1xpsixpsiz2r4" xml:space="preserve">[[Bally]]. [[Jump the Shark]] -&gt; [[Henry Winkler]] -&gt; [[The Lords of Flatbush]] -&gt; [[Sylvester Stallone]] -&gt; [[Judge Dredd]] -&gt; [[Bally]]. - [[User:Nunh-huh¦Nunh-huh]] 11:09, 12 Jun 2005 (UTC)
905 A character [[The Fonz¦Arthur &quot;Fonzie&quot; Fonzarelli]] on a popular 1970s TV show [[Happy Days]] did something [[jumping the shark¦jumped over a tank containing a shark while on water skis]] which fans later claimed signalled the end of the show [[1984]]. The actor [[Henry Winkler]] who played that character started his career in a movie [[The Lords of Flatbush]] with an actor [[Sylvester Stallone]] who went on to star in many films (surely you don't want all of these?), including one based on a comic strip [[Judge Dredd]]. Who made the pinball machine? [[Bally]] - [[User:Nunh-huh¦Nunh-huh]]
906 Norton started out as a professional [[arm wrestling¦arm wrestler]].&lt;ref name=Cage/&gt; He won over 30 championships during his days as an arm wrestler,&lt;ref name=Where1/&gt; including four U.S. National championships,&lt;ref name=&quot;Interview&quot;&gt;{{cite web¦url=http://slam.canoe.ca/Slam/Wrestling/2006/10/26/2144195.html¦archive-url=https://archive.today/20130115142041/http://slam.canoe.ca/Slam/Wrestling/2006/10/26/2144195.html¦url-status=usurped¦archive-date=January 15, 2013¦title=Scott Norton gives promoting a try¦access-date=13 February 2007¦author=Greg Oliver¦date=2006-10-26}}&lt;/ref&gt; and gained the nickname &quot;Flash&quot; due to his quickness in winning matches.&lt;ref name=Where1/&gt; His status in the arm wrestling world earned him a role as an extra in [[Sylvester Stallone]]'s arm wrestling movie, ''[[Over the Top (1987 film)¦Over the Top]]''.&lt;ref name=Where1/&gt;&lt;ref&gt;{{cite video¦people=Cast List¦title=Over the
```



```
    Top|url=https://www.imdb.com/title/tt0093692/|medium=Video| publisher=Cannon Group|location=USA|date=13 February 1987}}&lt;/ref&gt; While touring the arm wrestling circuit in Japan, Norton was approached by [[New Japan Pro-Wrestling]] about becoming a professional wrestler, but he turned it down initially.&lt;ref name=Where1/&gt;&lt;ref name=&quot;Interview&quot;/&gt; In addition to his professional arm wrestling career, Norton also worked as a bodyguard for [[Prince (musician)|Prince]] during the musician's [[1999 Tour|1999]]&lt;ref&gt;{{cite web|url=http://www.myspace.com/nortonsan|title=Scott Norton's myspace}}&lt;/ref&gt; and [[Purple Rain Tour]]s.&lt;ref name=Where1&gt;{{cite news|url=http://www.wwe.com/classics/wherearetheynow/where-are-they-now-scott-norton-26183170|title=Where Are They Now?: Scott Norton, page 1|newspaper=Wwe}}&lt;/ref&gt;
907 He appeared in the 1978 [[Sylvester Stallone]] movie ''[[Paradise Alley]]''.&lt;ref name=Encyclopedia2/&gt;
908 * Martinex appears in ''[[Guardians of the Galaxy Vol. 2]]'', portrayed by [[Michael Rosenbaum]].&lt;ref&gt;{{cite web|url=http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu|title=Sylvester Stallone's 'Guardians of the Galaxy Vol. 2' character has a big future in the MCU|last=Daniell|first=Mark|work=The Whig|date=April 25, 2017|access-date=April 28, 2017|archive-url=https://web.archive.org/web/20170428051803/http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu|archive-date=April 28, 2017|url-status=live}}&lt;/ref&gt; This version is a member of [[Stakar Ogord (Marvel Cinematic Universe)|Stakar Ogord]]'s [[Ravagers (Marvel Cinematic Universe)|Ravagers]]. He comes into conflict with [[Yondu Udonta (Marvel Cinematic Universe)|Yondu Udonta]] after he breaks the Ravagers' law against [[child trafficking]], but comes to respect him after he sacrifices himself to save [[Peter Quill (Marvel Cinematic Universe)|Peter Quill]] and helps reunite the Ravagers in his honor.
909 In 2001 Bräck also had a minor role in the [[Cinema of the United States|Hollywood]] [[motion picture]] ''[[Driven (2001 film)|Driven]]'', which starred [[Sylvester Stallone]] and [[Burt Reynolds]].
910 Shahi was cast as a recurring love interest for [[Taylor Kinney]]'s character Lt. Kelly Severide on [[NBC]]'s show ''[[Chicago Fire (TV series)|Chicago Fire]]'' in October 2012.&lt;ref&gt;{{cite web|url=http://tvline.com/2012/10/30/chicago-fire-season-1-cast-sarah-shahi/|title='Chicago Fire' Cast - Sarah Shahi to Play Taylor Kinney's Love Interest|date=October 30, 2012|publisher=[[TVLine]]|author=Michael Ausiello|access-date=December 28, 2012}}&lt;/ref&gt; She was cast in the recurring role of Sameen Shaw on ''[[Person of Interest (TV series)|Person of Interest]]'', making her first appearance in the second-season episode &quot;[[Relevance (Person of Interest)|Relevance]]&quot;, which aired in February 2013.&lt;ref&gt;{{cite magazine|url=https://ew.com/article/2013/01/08/sarah-shahi-to-recur-on-person-of-interest/|title=Sarah Shahi to recur on 'Person of Interest'|date=January 8, 2013|author=Sandra Gonzalez|magazine=Entertainment Weekly|access-date=February 21, 2013}}&lt;/ref&gt; In May 2013, CBS President [[Nina Tassler]] announced that Shahi would be added as a series regular for its third season.&lt;ref name=&quot;POI&quot;/&gt; She played the daughter of [[Sylvester Stallone]]'s character in the 2012 action film ''[[Bullet to the Head]]''. Her co-star on this film was [[Jason Momoa]], who she worked with again in the 2014 film ''[[Road to Paloma]]''. Following the January 7, 2015, episode of ''Person of Interest'', Shahi and the show's producers announced she would be going on maternity leave.&lt;ref&gt;{{cite web |last1=Bryant |first1=Adam |title=Person of Interest Bosses, Star on Shocking Exit and That Big Kiss |url=https://www.tvguide.com/news/person-of-interest-sarah-shahi-jonathan-nolan-1091505/ |website=TV Guide |access-date=April 6, 2022 |date=January 8, 2015|archive-url=https://web.archive.org/web/20150111103926/http://www.tvguide.com/News/Person-of-Interest-Sarah-Shahi-Jonathan-Nolan-1091505.aspx|archive-date=January 11, 2015}}&lt;/ref&gt;
911 * &quot;Mother&quot; was used in [[Sylvester Stallone]]'s sports drama ''[[Driven (2001 film)|Driven]]''.&lt;ref&gt;[http://www.soundtrack.net/album/driven/ &quot;''Driven'' Soundtrack on Soundtrack.net] {{Webarchive|url=https://web.archive.org/web/20160918025528/http://www.soundtrack.net/album/driven/ |date=2016-09-18 }}. Retrieved 4 September 2016.&lt;/ref&gt;
912 Other Montegrappa products were watches,&lt;ref&gt;{{cite web |url=http://www.montegrappa.com/watches.html |title=Elmo &amp; Montegrappa |accessdate=2014-02-21 |url-status=dead |archiveurl=https://web.archive.org/web/20140301201623/http://www.montegrappa.com/watches.html |archivedate=2014-03-01 }}&lt;/ref&gt; cufflinks,&lt;ref&gt;{{cite web |url=http://www.montegrappa.com/cuff-links.html |title=Elmo &amp; Montegrappa |accessdate=2014-02-21 |url-status=dead |archiveurl=https://web.archive.org/web/20140301205922/http://www.montegrappa.com/cuff-links.html |archivedate=2014-03-01 }}&lt;/ref&gt; pockets of leather, fragrances and accessories. The Montegrappa company is owned by Aquila's family: [[Sylvester Stallone]] and the ex-F1 Ferrari driver [[Jean Alesi]] are also partners and members of the board for P.R. and communication.&lt;ref&gt;
913 [http://ricerca.repubblica.it/repubblica/archivio/repubblica/2011/04/11/penne-montegrappa-venduta-ricomprata-ora-anche-stallone.html La Repubblica]&lt;/ref&gt;&lt;ref&gt;[http://www.ilgiornaledellusso.it/2013/05/31/che-chaos-di-orologio/ il giornale del lusso]&lt;/ref&gt;&lt;ref&gt;[http://corrieredelveneto.corriere.it/veneto/notizie/cronaca/2010/10-dicembre-2010/stallone-entra-montegrappa-firmera-penna-specialissima-18150494809.shtml Corriere del Veneto]&lt;/ref&gt;&lt;ref&gt;[http://www.lussuosissimo.com/sylvester-stallone-ambasciatore-del-marchio-montegrappa/ Sylvester Stallone ambasciatore del marchio Montegrappa]&lt;/ref&gt;
914 * [[Sylvester Stallone]]
915 '''''Tango &amp; Cash''''' is a 1989 American [[buddy cop film|buddy cop]] [[action comedy film]] starring [[Sylvester Stallone]], [[Kurt Russell]], [[Jack Palance]], and [[Teri Hatcher]]. The film follows the titular pair of rival police detectives who are forced to work together after a criminal mastermind frames them for murder.
916 * [[Sylvester Stallone]] as Lieutenant Raymond &quot;Ray&quot; Tango, the best cop in West Los Angeles. A slick, refined detective who trades stocks on the side, Tango wears three-piece [[Armani]] suits, drives a [[Cadillac Allanté]], carries a [[Smith &amp; Wesson Model 36]] as his sidearm, and lives in a middle-class house with his sister.
917 The film was known as ''The Set Up'' and was based on a script by Randy Feldman which was based on an idea by [[Jon Peters]] and [[Peter Guber]]. [[Sylvester Stallone]] and [[Patrick Swayze]] were signed to star.&lt;ref name=&quot;AFI film-58307&quot; /&gt;&lt;ref&gt;{{cite news |last=Klady |first=Leonard |date=12 February 1989 |title=Cinefile: [Home Edition] |page=35 |newspaper=Los Angeles Times}}&lt;/ref&gt; In March 1989 [[Andrei Konchalovsky]] signed to direct.&lt;ref&gt;{{cite news|title=Cinefile: [Home Edition]
918 Sylvester Stallone had the original director of photography, [[Barry Sonnenfeld]], fired.&lt;ref name=Dissolve/&gt; [[Donald E. Thorin]], who had shot Stallone's movie ''[[Lock Up (1989 film)|Lock Up]]'' earlier that year, was Sonnenfeld's replacement.&lt;ref&gt;{{cite news |title=Barry Sonnenfeld: Memoirs of a Neurotic Filmmaker |url=https://thescriptblog.com/barry-sonnenfeld-memoirs-of-a-neurotic-filmmaker/ |access-date=14 December 2023 |publisher=The Script Blog |date=17 May 2023}}&lt;/ref&gt;
919 James agreed that the official reason Konchalovsky was fired was because of the budget, but he also said that going over budget was not Konchalovsky's fault, and that Konchalovsky did not deserve to be fired. Konchalovsky, however, had nothing but praise for Sylvester Stallone, and both he and James said that despite Stallone's ego and decision to fire the original cinematographer, and the fact that he had a hand in Konchalovsky's firing, Stallone was the one person who held the project together, and that he was a constant voice of reason on an increasingly chaotic set. According to Konchalovsky, by the end of principal photography, Stallone was unofficially working as producer, director and writer, as well as star, and Konchalovsky believes that had it not been for Stallone, Peters would have fired him much sooner than he did. Production sources said that Konchalovsky had been given impossible scheduling demands, and was then made the scapegoat when he fell behind.&lt;ref name=&quot;brionjames-reocities&quot;&gt;{{cite web |last=Paul |first=Louis |date=August 2009 |title=Interview with Brion James |url=http://www.reocities.com/Hollywood/movie/2919/interview.html |url-status=dead |archive-url=https://web.archive.org/web/20170410095744/http://www.reocities.com/Hollywood/Movie/2919/interview.html |archive-date=10 April 2017 |access-date=24 August 2016 |website=reocities.com}}&lt;/ref&gt;
920 ''Tango &amp; Cash'' was nominated for three [[Golden Raspberry Awards]] for Worst Actor (Sylvester Stallone), Worst Supporting Actress (Kurt Russell in drag) and Worst Screenplay.&lt;ref name=&quot;Rasp&quot;&gt;{{cite web |title=1989 Archive |url=http://www.razzies.com/asp/content/XcNewsPlus.asp?cmd=view&amp;articleid=29 |website=razzies.com |access-date=10 February 2023 |archive-url=https://web.archive.org/web/20120707044333/http://www.razzies.com/asp/content/XcNewsPlus.asp?cmd=view&amp;articleid=29 |archive-date=July 7, 2012}}&lt;/ref&gt;
921 In September 2019, Stallone revealed that he had a story prepared for a potential sequel and was trying to convince Kurt Russell to sign onto the project. However, while Stallone was excited to film a sequel, Russell was unsure if he wanted to, saying that by then they were in their &quot;unprime&quot;.&lt;ref&gt;{{Cite web|url=https://www.fandango.com/movie-news/sylvester-stallone-teaming-with-robert-rodriguez-on-cobra-reboot-is-tango-and-cash-next-753897|title=Sylvester Stallone Teaming with Robert Rodriguez on 'Cobra' Reboot; Is 'Tango &amp; Cash' Next? |date=September 16, 2019 |access-date=2019-09-16|archive-date=2020-06-01|archive-url=https://web.archive.org/web/20200601133251/https://www.fandango.com/movie-news/sylvester-stallone-teaming-with-robert-rodriguez-on-cobra-reboot-is-tango-and-cash-next-753897|url-status=live|website=[[Fandango Media|Fandango]]}}&lt;/ref&gt; As of 2023, no further news has materialized on the sequel.
922 * [[Sylvester Stallone]]
923 '''''Stop! Or My Mom Will Shoot''''' is a 1992 American [[buddy cop]] [[action comedy]] directed by [[Roger Spottiswoode]] and starring [[Sylvester Stallone]] and [[Estelle Getty]]. The film was released in the United States on February 21, 1992. The film received highly negative reviews upon release but grossed $70.6 million worldwide.
924 * [[Sylvester Stallone]] as Sergeant Joseph Andrew &quot;Joe&quot; Bomowski
925 Rita Kempley of ''[[The Washington Post]]'' called it &quot;[[q:Rambo III#Dialogue|your worst nightmare]]&quot; but stated that &quot;the concept is actually better for Stallone than the premises of his earlier awful romps, ''[[Rhinestone (film)|Rhinestone]]'' and ''[[Oscar (1991 film)|Oscar]].''&quot;&lt;ref&gt;{{cite news|url=https://www.washingtonpost.com/wp-srv/style/longterm/movies/videos/stopormymomwillshootpg13kempley_a0a29c.htm|title='Stop! Or My Mom Will Shoot' (PG-13)|last=Kempley|first=Rita|date=February 21, 1992|newspaper=[[The Washington Post]]|access-date=September 10, 2010}}&lt;/ref&gt; Clifford Terry wrote in the ''[[Chicago Tribune]]'' that the film &quot;plays like an extended sitcom-perhaps four episodes of ''[[She's the Sheriff]]''&quot; and also that &quot;About two-thirds into ''Stop! Or My Mom Will Shoot,'' Sylvester Stallone actually delivers the title line. That's how numbingly awful this is. Give it half a star for being in focus.&quot;&lt;ref&gt;{{cite news|url=https://www.chicagotribune.com/1992/02/21/stop-or-my-mom-will-shoot-misses-the-mark-by-a-mile/|title='STOP! OR MY MOM WILL SHOOT' MISSES THE MARK BY A MILE|last=Terry|first=Clifford|date=February 21, 1992|work=[[Chicago Tribune]]|access-date=September 10, 2010}}&lt;/ref&gt;  Michael Wilmington of the ''[[Los Angeles Times]]'' wrote that the film seemed like Stallone's response to Schwarzenegger's turn to comedies like ''[[Kindergarten Cop]]'' and added, &quot;This is another 'high-concept' marketing hook job-a slick, slow-witted, shiny, 100% predictable movie-and the scriptwriters ... don't have anything richer on their minds than the usual feisty mother-son gags.&quot;&lt;ref&gt;Wilmington, Michael (February 21, 1992). &quot;'Stop!': Stallone KOs His Macho Image&quot;. ''[[Los Angeles Times]]''. F14.&lt;/ref&gt;&lt;ref&gt;{{cite web |title=MOVIE REVIEW : 'Stop!': Stallone KOs His Macho Image |url=https://www.latimes.com/archives/la-xpm-1992-02-21-ca-2433-story.html |website=Los Angeles Times |date=21 February 1992 }}&lt;/ref&gt;
926 Both [[Gene Siskel]] and [[Roger Ebert]] disliked the film, and both gave it a thumbs down in their onscreen review of the film. Siskel said &quot;If this doesn't turn out to be one of the very worst movies of the year, it's gonna be a VERY bad year,&quot; while Ebert called it &quot;one of the worst movies I've ever seen&quot;; in his newspaper review (in which he awarded half of one star out of four), Ebert wrote while Stallone and Getty had both performed well in other comedic roles, ''Stop! Or My Mom Will Shoot'' was &quot;one of those movies so dimwitted, so utterly lacking in even the smallest morsel of redeeming value, that you stare at the screen in stunned disbelief. It is moronic beyond comprehension, an exercise in desperation during which even Sylvester Stallone, a repository of self-confidence, seems to be disheartened.&quot;&lt;ref&gt;{{cite web |last=Ebert |first=Roger |author-link=Roger Ebert |title=Stop! Or My Mom Will Shoot movie review (1992) |url=https://www.rogerebert.com/reviews/stop-or-my-mom-will-shoot-1992 |website=[[Chicago Sun-Times]] }}&lt;/ref&gt; Siskel gave the film zero stars out of four and stated that if the script had been submitted to the staff of ''[[The Golden Girls]]'', which co-starred Getty, it &quot;would be summarily dismissed as too flimsy for a half-hour sitcom. There is not one laugh nor surprising moment to be found, starting with the scene where Stallone and Getty happen upon a jumper atop a building and Getty manages to bring the man down safely using a bullhorn.&quot;&lt;ref&gt;{{cite news |url=https://www.chicagotribune.com/1992/02/21/stallones-unfunny-stop-shoots-down-creativity/ |title=STALLONE'S UNFUNNY 'STOP!' SHOOTS DOWN CREATIVITY |last=Siskel |first=Gene |author-link=Gene Siskel |date=February 21, 1992 |work=[[Chicago Tribune]] |access-date=December 12, 2010|url-status=live |archive-url=https://web.archive.org/web/20120913141443/http://articles.chicagotribune.com/1992-02-21/entertainment/9201170069_1_stallone-and-getty-water-tower-kindergarten-cop|archive-date=September 13, 2012}}&lt;/ref&gt;
927 Sylvester Stallone has stated that ''Stop! Or My Mom Will Shoot'' was the worst film he had ever starred in. He told [[Ain't It Cool News]] that it was &quot;maybe one of the worst films in the
```

entire solar system, including alien productions we've never seen&quot;, that &quot;a [[flatworm]] could write a better script&quot;, and that &quot;in some countries - China, I believe - running [the movie] once a week on government television has lowered the birth rate to zero. If they ran it twice a week, I believe in twenty years China would be extinct.&quot;&lt;ref name=&quot;aintitcool-dec4&quot;&gt;{{cite web |url=http://www.aintitcool.com/node/30869 |title=Round Three!! Dec 3rd's Sly answers to your Stallone'd Questions |author=headgeek |date=December 4, 2006 |website=[[Ain't It Cool News]] |access-date=August 12, 2012 }}&lt;/ref&gt;&lt;ref name=&quot;aintitcool-dec6&quot;&gt;{{Cite news|url=http://www.aintitcool.com/node/30899|title=Round #5 - Stallone keeps slugging out answers to the AICN Mob!!!|last=headgeek|date=December 6, 2006|work=[[Ain't It Cool News]]|access-date=March 19, 2018 }}&lt;/ref&gt;

928 Later Stallone said:&lt;blockquote&gt; Stop! Or My Mom Will Shoot was supposed to be like [[Throw Momma from the Train|''Throw Momma From the Train'']] with the mom as this really nasty piece of work. Instead you hire the nicest woman in Hollywood, Estelle Getty, who you wish was your mother. That's the end of that! Also, I had heard Schwarzenegger was going to do that movie and I said, &quot;I'm going to beat him to it.&quot; I think he set me up.&lt;ref name=&quot;:0&quot;&gt;{{cite web|url=https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/|title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot;|work=The Hollywood Reporter|first=James|last=Hibberd|date=November 7, 2022|access-date=November 29, 2022}}&lt;/ref&gt;&lt;/blockquote&gt;

929 The film was mentioned when Stallone hosted an episode of ''[[Saturday Night Live]]'' in 1997; in one particular skit, Stallone comes across someone in a terrible car accident ([[Norm Macdonald]]) who does not like any of his work and ridicules his films. As he lies dying, he mutters something quietly that only Stallone can hear, and when a passerby ([[Will Ferrell]]) asks what he said, Stallone is reluctant to say it until he is grilled some more, at which point he virulently yells &quot;He said ''Stop! Or My Mom Will Shoot''...SUCKED!&quot;&lt;ref&gt;{{cite web |title=Watch Stallone Can't Save Everything From Saturday Night Live - NBC.com |url=http://www.nbc.com/saturday-night-live/video/bad-stallone-movies/n11010?snl=1 |archive-url=https://web.archive.org/web/20170430100542/http://www.nbc.com/saturday-night-live/video/bad-stallone-movies/n11010?snl=1 |archive-date=2017-04-30 |quote=Sylvester Stallone tries to help a couple (Norm Macdonald, Ana Gasteyer) after they get in a horrible car accident, but the two can't stop insulting Stallone and his movies despite being in pain. [Season 23, 1997] |url-status=dead}}&lt;/ref&gt;

930 In ''[[Mortal Kombat 11]]'', during a pre-match dialogue exchange between [[John Rambo]] (voiced by Stallone) and [[Cassie Cage]], Cassie references the film's title.&lt;ref&gt;[https://www.youtube.com/watch?v=1YXhyQJEmE8 MK11 Rambo All Sylvester Stallone Movie Easter Eggs References Mortal Kombat 11 by MKIceAndFire on YouTube]&lt;/ref&gt;

931 * June 22 - The movie ''[[Rhinestone (film)|Rhinestone]]'', starring [[Dolly Parton]] and [[Sylvester Stallone]], is released to universally negative reviews. The much-hyped movie - about a singer's effort to transform a New York City taxicab driver into a country star within two weeks - flops, but still produces several hit singles, most notably the No. 1 hit &quot;[[Tennessee Homesick Blues]]&quot;.

932 It was announced in August 2008 that he would write [[Dynamite Entertainment]]'s series ''The Man with No Name'' based on the [[Man with No Name|Western character]].&lt;ref&gt;{{cite web |url= http://www.newsarama.com/826-chuck-dixon-to-write-the-man-with-no-name.html|title= Chuck Dixon to Write ''The Man With No Name''|first= Matt|last= Brady|date= August 20, 2008|publisher= [[Newsarama]]|archive-url= https://web.archive.org/web/20140219200846/http://www.newsarama.com/826-chuck-dixon-to-write-the-man-with-no-name.html|archive-date= February 19, 2014|url-status= live|df=mdy-all|access-date= September 28, 2013}}&lt;/ref&gt; He wrote a ''[[G.I. Joe]]'' series for [[IDW Publishing]].&lt;ref&gt;{{cite web|url= http://www.comicbookresources.com/?page=article&amp;id=17962|title= Chuck Dixon Writes ''G.I. Joe'' for IDW|first= Van|last= Jensen|date= September 8, 2008|publisher= Comic Book Resources|archive-url= https://web.archive.org/web/20130717180131/http://www.comicbookresources.com/?page=article&amp;id=17962|archive-date= July 17, 2013|url-status= dead|df= mdy-all|access-date= September 28, 2013}}&lt;/ref&gt; In March 2009 [[Moonstone Books]] published a new [[Airboy]] one-shot written by Dixon entitled ''Airboy 1942: The Best of Enemies.'' In 2011, Dixon says he was offered a chance to do a rewrite on ''[[The Expendables 2]]'' screenplay by Sylvester Stallone, but declined.&lt;ref&gt;{{cite web |url= http://dangapotamus.com/DoubleDang/2013/07/01/277/|title= Q&amp;A with Chuck Dixon|date= July 1, 2013|publisher= Dangapotamus.com|archive-url= https://web.archive.org/web/20130929044419/http://dangapotamus.com/DoubleDang/2013/07/01/277/|archive-date= September 29, 2013|url-status= dead|df=mdy-all}}&lt;/ref&gt;

933 [[Roger Corman]] gave Bartel the job of directing ''[[Death Race 2000]]'' (1975), a satirical action comedy starring [[David Carradine]], [[Sylvester Stallone]] and [[Mary Woronov]]. Bartel also played a small role. The film was a huge success at the box office and quickly established itself as a cult favorite.

934 |[[Sylvester Stallone]]

935 Billy Bob's Texas has also been host to many movie and television projects. ''[[Baja Oklahoma]]'' (Willie Nelson, Lesley Ann Warren), ''[[Over the Top (1987 film)|Over the Top]]'' (Sylvester Stallone), and ''[[Necessary Roughness (film)|Necessary Roughness]]'' (Scott Bakula, Sinbad) have all been filmed there.&lt;ref&gt;{{cite web|title=Dallas Film Commission - Billy Bob's Texas|url=http://www.dallasfilmcommission.com/film-tourism/?doing_wp_cron=1503680219.6527678966522216796875/}}&lt;/ref&gt;

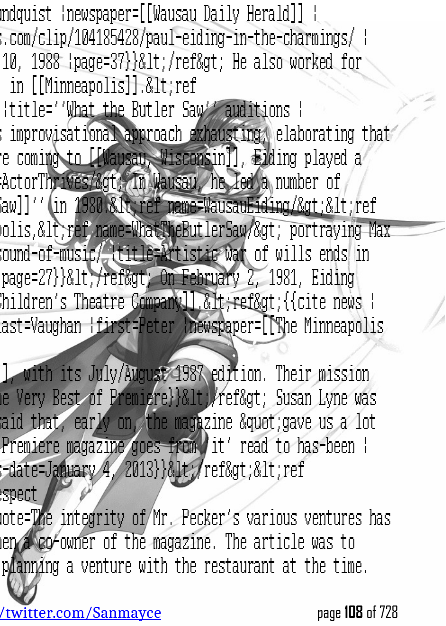
936 Kiniski went into acting on occasion, appearing in the 1978 [[Sylvester Stallone]] movie ''[[Paradise Alley]]'', as well as ''[[Double Happiness (film)|Double Happiness]]'' (which also starred [[Sandra Oh]]) and the made-in-Vancouver cult film ''[[Terminal City Ricochet]]'' (with former ''[[The Big Valley|Big Valley]]'' regular [[Peter Breck]]), playing a bit role as a policeman. On the TV side, he appeared as one of the wrestling legends interviewed on [[The Comedy Network]] series ''Wrestling with the Past'', and he served as a commercial pitchman in several TV ads aired in the Vancouver area.&lt;ref&gt;{{YouTube|oN-VF9aFNII|Gene Kiniski commercial for Parker's Mattress City}}&lt;/ref&gt; He also made an appearance on an episode of the [[CBUT-DT|CBUT]] arts and entertainment magazine show ''Zero Avenue'' in 1993, discussing art with host/interviewer Christine Lippa in a Vancouver-area art gallery.

937 Along with some of his Ipswich teammates,&lt;ref&gt;{{cite web |url=https://www.eadt.co.uk/sport/town-players-in-escape-to-victory-reunion-1-835075 |work=[[East Anglian Daily Times]] |title=Town players in 'Escape to Victory' reunion |first=Elvin |last=King |date=23 March 2011 |access-date=5 December 2018 |archive-url=https://web.archive.org/web/20181205193512/https://www.eadt.co.uk/sport/town-players-in-escape-to-victory-reunion-1-835075 |archive-date=5 December 2018 |url-status=live |df=dmy-all}}&lt;/ref&gt; Beattie featured in the 1981 film ''[[Escape to Victory]]''. His skills were shown on the pitch as the [[body double]] for [[Michael Caine]]'s prisoner-of-war character, and the two became friends.&lt;ref name=Times&gt;{{cite news |url=https://www.thetimes.co.uk/article/beattie-the-best-england-player-of-his-generation-dies-at-64-tvz0bkmwg |title=Beattie, 'the best England player of his generation', dies at 64 |first=George |last=Caulkin |newspaper=[[The Times]] |access-date=17 September 2018 |archive-url=https://web.archive.org/web/20181019005349/https://www.thetimes.co.uk/article/beattie-the-best-england-player-of-his-generation-dies-at-64-tvz0bkmwg |archive-date=19 October 2018}}&lt;/ref&gt; Beattie had a cooler relationship with [[Sylvester Stallone]], who also starred in the film: &quot;There weren't too many that got on that well with him and after I beat him in an arm wrestle&amp;nbsp;- first my right arm and then my left - he didn't speak to me again&quot;.&lt;ref name=Times/&gt; A campaign for a permanent memorial to Beattie to be placed outside Portman Road, along with the existing statues of Robson and Ramsey, was started soon after his death by ''[[Ipswich Star]]'' and ''[[East Anglian Daily Times]]'' editor Brad Jones.&lt;ref&gt;{{cite news |url=https://www.bbc.co.uk/news/uk-england-suffolk-45560876 |work=[[BBC News]] |title=Kevin Beattie: Calls for statue of 'sensational player' |access-date=21 October 2018 |archive-url=https://web.archive.org/web/20181018012814/https://www.bbc.co.uk/news/uk-england-suffolk-45560876 |archive-date=18 October 2018 |url-status=live |df=dmy-all |date=18 September 2018}}&lt;/ref&gt; The design for the statue was revealed on 19 March 2019 by local sculptor [[Sean Hedges-Quinn]].&lt;ref&gt;{{cite web |url=https://www.bbc.co.uk/news/uk-england-suffolk-47614509 |work=BBC News |date=19 March 2019 |title=Ipswich Town: Kevin Beattie statue design revealed |access-date=13 September 2019}}&lt;/ref&gt; It was confirmed in August 2019 that the funds required to build the statue had been raised.&lt;ref&gt;{{cite web |url=https://www.eadt.co.uk/news/kevin-beattie-statue-appeal-hits-target-1-6236257 |work=East Anglian Daily Times |title=You've done it! Kevin Beattie statue appeal hits target thanks to Blues fans |first=Brad |last=Jones |access-date=13 September 2019 |date=27 August 2019 |archive-date=15 September 2019 |archive-url=https://web.archive.org/web/20190915173822/https://www.eadt.co.uk/news/kevin-beattie-statue-appeal-hits-target-1-6236257 |url-status=dead }}&lt;/ref&gt; On 18 December 2021, on what would have been Beattie's 68th birthday, his statue was officially unveiled outside Portman Road.&lt;ref&gt;{{cite news |url=https://www.bbc.com/news/uk-england-suffolk-59710167 |title=Ipswich Town's Kevin Beattie statue unveiled at Portman Road |date=18 December 2021 |work=BBC News |access-date=10 January 2022}}&lt;/ref&gt;

938 | [[Walter Hill (director)|Walter Hill]]&lt;ref&gt;{{Cite news|last=Kilday|first=Gregg|title=Joel Silver's Dark Castle Boards Sylvester Stallone Action Film|url=https://www.hollywoodreporter.com/news/general-news/joel-silvers-dark-castle-boards-196950/|date=June 10, 2011|website=Hollywood Reporter}}&lt;/ref&gt;

939 During the 1970s, Eiding was a prolific commercial actor, appearing in advertisements for snowmobiles, granola bars, department stores, and clothing.&lt;ref name=ActorThrives&gt;{{cite news |url=https://www.newspapers.com/clip/119568769/actor-thrives-on-theater-of-today/ |title=Actor thrives on theater of today |first=Barbara |last=Lundquist |newspaper=[[Wausau Daily Herald]] |location=[[Wausau, Wisconsin]] |date=September 9, 1977 |page=24}}&lt;/ref&gt;&lt;ref name=WausauEiding&gt;{{cite news |url=https://www.newspapers.com/clip/104185428/paul-eiding-in-the-charmings/ |title=Features: Town &amp; Country |last=Williams |first=Joanne |newspaper=[[Wausau Daily Herald]] |location=[[Wausau, Wisconsin]] |date=January 10, 1988 |page=37}}&lt;/ref&gt; He also worked for theater groups across the [[Midwestern United States|Midwest]],&lt;ref name=WausauEiding/&gt; including [[Dudley Riggs]]'s [[Brave New Workshop]] in [[Minneapolis]].&lt;ref name=ActorThrives/&gt;&lt;ref name=WhatTheButlerSaw&gt;{{cite news |url=https://www.newspapers.com/clip/104187760/what-the-butler-saw-auditions/ |title=''What the Butler Saw'' auditions |newspaper=[[Wausau Daily Herald]] |location=[[Wausau, Wisconsin]] |date=May 23, 1980 |page=19}}&lt;/ref&gt; Eiding found the Brave New Workshop's improvisational approach exhausting, elaborating that &quot;You have to take time off to think of new ideas. At Dudley Riggs we were doing nine shows a week&quot;.&lt;ref name=ActorThrives/&gt; Before coming to [[Wausau, Wisconsin]], Eiding played a small part aside [[Sylvester Stallone]] in the [[Norman Jewison]] film ''[[F.I.S.T. (film)|F.I.S.T.]]'', which was shot in [[Iowa]].&lt;ref name=ActorThrives/&gt; In Wausau, he led a number of workshops, and was a guest director for its community theater's production of the comedy-mystery ''[[What the Butler Saw (play)|What the Butler Saw]]'' in 1980.&lt;ref name=WausauEiding/&gt;&lt;ref name=WhatTheButlerSaw/&gt; He was a director and actor for the [[Dinner theater#Chanhassen Dinner Theatres|Chanhassen Dinner Theater]] of Minneapolis,&lt;ref name=WhatTheButlerSaw/&gt; portraying Max Detweiler in its 1981 production of ''[[The Sound of Music]]''.&lt;ref&gt;{{cite news |url=https://www.newspapers.com/clip/104191219/chanhassen-sound-of-music/ |title=Artistic war of wills ends in ''Sound of Music'' |last=Vaughan |first=Peter |newspaper=[[The Minneapolis Star]] |location=[[Minneapolis]], [[Minnesota]] |date=March 20, 1981 |page=27}}&lt;/ref&gt; On February 2, 1981, Eiding hosted the [[Minneapolis-Saint Paul|Twin Cities]]' first award ceremony for distinguished achievement in local theater, which was held at the [[Children's Theatre Company]].&lt;ref&gt;{{cite news |url=https://www.newspapers.com/clip/119572791/childrens-theatres-500-hats-is-big/ |title=Children's Theatre's ''500 Hats'' is big Kudos winner |last=Vaughan |first=Peter |newspaper=[[The Minneapolis Star]] |location=[[Minneapolis]], [[Minnesota]] |date=February 3, 1981 |page=15}}&lt;/ref&gt;

940 The U.S. version of the magazine was launched by [[News Corporation (1980-2013)|News Corporation]], based in [[New York City]] and [[Los Angeles]], with its July/August 1987 edition. Their mission was to &quot;reflect The Second Golden Age of the Movies&quot;.&lt;ref&gt;{{cite magazine|magazine=Premiere|date=Autumn 1992|pages=70-79|title=The Very Best of Premiere}}&lt;/ref&gt; Susan Lyne was the founding editor, and among those working for the magazine was [[Peter Biskind]], who spent a decade at the magazine as executive editor. He said that, early on, the magazine &quot;gave us a lot of freedom to do hard-hitting, in-depth reporting.&quot;&lt;ref name=&quot;LA&quot;&gt;{{cite news | last = Welkos | first = Robert W. | title = Premiere magazine goes from 'it' read to has-been | work =[[Los Angeles Times]] | date = March 10, 2007 | url = https://www.latimes.com/archives/la-xpm-2007-mar-10-et-premiere10-story.html | access-date=January 4, 2013}}&lt;/ref&gt;&lt;ref name=&quot;renegade&quot;&gt;{{cite news |date=April 6, 1998 |first=Robin |last=Pogrebin |title=Once a Renegade, Hachette Magazine Chief Gains Respect |url=https://www.nytimes.com/1998/04/06/business/once-a-renegade-hachette-magazine-chief-gains-respect.html |newspaper=[[The New York Times]] |quote=The integrity of Mr. Pecker's various ventures has been questioned over the years, particularly after he killed an article for ''Premiere'' on behalf of the Revlon executive Ronald O. Perelman, then a co-owner of the magazine. The article was to examine the business dealings of the actor [[Sylvester Stallone]], including his role in the [[Planet Hollywood]] restaurant chain. Perelman was planning a venture with the restaurant at the time. Premiere's two top editors resigned over the incident.}}&lt;/ref&gt;



941 In December 2016, the Bureau of Engineering once again recommended razing the building, saying that tearing down the structure and creating the new high-rise would cost $514 million, and that the preservation and smaller new edifice option would cost $621 million (both plans would have roughly the same amount of office space). The Bureau of Engineering's timetable suggested securing approvals by February 2017.&lt;ref&gt;{{cite news¦ title=City Recommends Razing Parker Center, Again¦ first=Eddie¦ last=Kim¦ date=December 16, 2016¦ url=http://www.ladowntownnews.com/news/city-recommends-razing-parker-center-again/article_2a56ada0-c330-11e6-afec-d37da0848377.html¦ newspaper=Los Angeles Downtown News}}&lt;/ref&gt;&lt;ref&gt;{{cite news¦ title=Roundup: Sylvester Stallone and NEA, art and Inauguration Day, future of Parker Center¦ url=https://www.latimes.com/entertainment/arts/miranda/la-et-cam-roundup-sylvester-stallone-parker-center-20161218-story.html¦ first=Carolina A.¦ last=Miranda¦ date=December 19, 2016¦ newspaper=Los Angeles Times¦ url-access=subscription}}&lt;/ref&gt;

942 An English-language version of the movie was made in 1991, by [[John Landis]], under the [[Oscar (1991 film)¦same name]] and starring [[Sylvester Stallone]].

943 *'''Sirloin:''' A bull and one of Croco's goons, with a voice-based on [[Sylvester Stallone]].

944 * [[Sylvester Stallone]] - Mafioso #2 (uncredited)

945 The cameo by [[Sylvester Stallone]] is uncredited, while [[Roger Corman]] and [[Don Simpson]] appear as [[district attorney]]s. Directors [[Joe Dante]], [[Jonathan Kaplan]], [[Allan Arkush]] and [[Martin Scorsese]] have cameos, and former beach movie star and [[Transformers]] voice actor [[Aron Kincaid]] appears in a small role as one of two cops who pull over the girls in the van.

946 * The Denver Mint is featured in the 1993 [[Sylvester Stallone]] film ''[[Cliffhanger (film)¦Cliffhanger]]'', as the production point of the money stolen in the film, and the departure point for the plane.

947 The Rock Island area was used for a considerable number of exterior shots and stunts in the 1994 [[Sylvester Stallone]] [[film]], ''[[The Specialist]]''.

948 Howard made his movie debut in 1970, in ''[[Tell Me That You Love Me, Junie Moon]]'', opposite [[Liza Minnelli]]. Numerous dramatic and comedic movie roles followed, including: [[Otto Preminger]]'s ''[[Such Good Friends]]'' with [[Dyan Cannon]] and [[Jennifer O'Neill]], ''[[1776 (film)¦1776]]'', ''[[The Strange Vengeance of Rosalie]]'', ''Independence'', ''[[Second Thoughts (1983 film)¦Second Thoughts]]'', ''[[Oscar (1991 film)¦Oscar]]'' with [[Sylvester Stallone]] in 1991, ''Ulterior Motives'', ''[[Clear and Present Danger (film)¦Clear and Present Danger]]'' with [[Harrison Ford]] in 1994, ''[[The Net (1995 film)¦The Net]]'' with [[Sandra Bullock]] in 1995, ''Tactical Assault'', ''[[Dreamer: Inspired by a True Story]]'', and ''[[In Her Shoes (film)¦In Her Shoes]]'' in 2005.

949 Silliphant wrote or co-wrote 47 feature films, including ''[[Maracaibo (film)¦Maracaibo]]'' (produced and directed by and starring [[Cornel Wilde]]); the Jacques Tourneur noir ''[[Nightfall (1957 film)¦Nightfall]]''; ''[[Village of the Damned (1960 film)¦Village of the Damned]]''; the [[Charles Bronson]] spy thriller ''[[Telefon (film)¦Telefon]]''; ''[[The Liberation of L.B. Jones]]'' (director [[William Wyler]]'s final film); ''[[The Killer Elite]]'' (directed by [[Sam Peckinpah]]); the [[Dirty Harry]] crime drama ''[[The Enforcer (1976 film)¦The Enforcer]]''; ''[[The Towering Inferno]]'', nominated for the Academy Award for Best Picture (1974); and the [[arm wrestling]] story ''[[Over the Top (1987 film)¦Over the Top]]'' (the latter with its star [[Sylvester Stallone]]).

950 Born in [[Nashville, Tennessee]], Rhodes began her show business career working at [[Opryland USA]] as a singer and dancer while attending [[Glencliff High School]]{{citation needed¦date=September 2019¦reason=Need to cite a reliable source to verify attendance at this school.}} during the 1970s.&lt;ref name=&quot;actor bio&quot;&gt;{{cite web ¦ url=https://movies.yahoo.com/movie/contributor/1800113431 ¦ website=Yahoo! Movies ¦ via=movies.yahoo.com ¦ publisher=[[Yahoo!]] ¦ title=Cynthia Rhodes Actor Biography ¦ access-date=January 5, 2008 ¦ archive-url=https://web.archive.org/web/20070422104006/http://movies.yahoo.com/movie/contributor/1800113431/bio ¦ archive-date=April 22, 2007 ¦ url-status=dead }}&lt;/ref&gt; Raised in a [[Baptist]] family, Rhodes tried to maintain a clean-cut image in her acting roles and in the media, turning down scripts that required nudity and refusing offers to pose for pictorials in ''[[Playboy]]'' magazine. Sylvester Stallone, the director of ''Staying Alive'', stated that Rhodes &quot;would sooner quit the business before doing anything to embarrass her parents.&quot;&lt;ref name=&quot;Magazine Article&quot;&gt;{{Cite web ¦ url = http://www.nctc.net/hazard/marx/cynthia/rosanna.html ¦ publisher=People Magazine/AOL Time Warner ¦ title=Two for Travolta ¦ via = www.nctc.net ¦ access-date=July 30, 2007 ¦ archive-url = https://web.archive.org/web/20070927203457/http://www.nctc.net/hazard/marx/cynthia/rosanna.html ¦ archive-date = September 27, 2007 ¦ url-status = dead}}&lt;/ref&gt;

951 Rhodes played a small role in the fantasy musical ''[[Xanadu (film)¦Xanadu]]'' (1980).&lt;ref name=&quot;film credits&quot;&gt;{{cite web ¦ url=http://www.hollywood.com/celebrity/Cynthia_Rhodes/190895 ¦ publisher=Hollywood.com ¦ title=Cynthia Rhodes Bio ¦ accessdate=July 30, 2007}}&lt;/ref&gt; In 1982 she appeared in a video production called &quot;The Tubes Video&quot; directed by [[Russell Mulcahy]] and choreographed by [[Kenny Ortega]]. This 52-minute long-form music video featured Rhodes as one of three backup dancers for the San Francisco-based rock band [[The Tubes]] and includes hits such as &quot;Talk to You Later&quot;, &quot;Sushi Girl&quot;, &quot;Sports Fans&quot; and &quot;Mondo Bondage&quot;.&lt;ref&gt;{{Citation¦title=The Tubes Video (1981) (SP 1982 Music Video VHS Rip)¦url=https://archive.org/details/thetubesvideo1981sp1982hifistereoemimusicvideovhsrip¦access-date=2020-09-12}}&lt;/ref&gt; Her next role was as Tina Tech in the musical film ''[[Flashdance]]''. After ''Flashdance'', Rhodes was cast opposite [[John Travolta]] in [[Sylvester Stallone]]'s 1983 film ''[[Staying Alive (1983 film)¦Staying Alive]]'', a sequel to the 1977 hit film ''[[Saturday Night Fever]]''. Rhodes's character, Jackie, was an ensemble dancer, bar band singer, and sometime love interest of Travolta's character. While poorly reviewed, the film was commercially successful.&lt;ref name=&quot;article&quot;&gt;{{cite magazine ¦ url=http://www.ew.com/ew/gallery/0,,20041669_20041686_1169126_25,00.html ¦ magazine=[[Entertainment Weekly]] ¦ title=25 Worst Sequels Ever Made ¦ access-date=February 11, 2009 ¦ archive-date=September 18, 2008 ¦ archive-url=https://web.archive.org/web/20080918080424/http://www.ew.com/ew/gallery/0,,20041669_20041686_1169126_25,00.html ¦ url-status=dead }}&lt;/ref&gt;

952 ''Dus'' is a tribute to celebrated director Mukul S. Anand, who had died while filming the 1997 incomplete film of the same name, and which had also starred [[Sanjay Dutt]] alongside [[Salman Khan]]. The film had a plot involving [[terrorism in Kashmir]], along with [[Sylvester Stallone]]-style action scenes. The film had a then unprecedented budget of {{INR¦200 million}}.&lt;ref&gt;{{cite news ¦ last=Chopra ¦first=Anupama ¦authorlink=Anupama Chopra ¦title=The Great Gamblers ¦url=https://www.indiatoday.in/magazine/indiascope/story/19961215-the-great-gamblers-753776-1996-12-15 ¦work=[[India Today]] ¦date=15 December 1996 ¦access-date=15 January 2019 ¦archive-date=16 January 2019 ¦archive-url=https://web.archive.org/web/20190116050148/https://www.indiatoday.in/magazine/indiascope/story/19961215-the-great-gamblers-753776-1996-12-15 ¦url-status=live }}&lt;/ref&gt; However, the film was cancelled after Anand's death. The director, [[Anubhav Sinha]] said the only similarity between the 1997 film and his film were only sharing the producer, Nitin Manmohan.

953 Pozniak won the prestigious Voice Arts Award in the &quot;Outstanding Video Game Character - Best Performance&quot; category for her role of Skarlet, the Blood Queen in [[Mortal Kombat 11]] starring alongside [[Arnold Schwarzenegger]] and [[Sylvester Stallone]]. In thirty years of [[Mortal Kombat]]'s franchise, Pozniak is the first and only actress that won in the Best Performance Category.&lt;ref&gt;{{cite web¦url=https://www.youtube.com/watch?v=7x4hj94KiBc&amp;t=2303s¦title=Kamidogu Podcast¦website=www.youtube.com¦date=1 March 2023 ¦accessdate=1 March 2023}}&lt;/ref&gt;

954 * [[Sylvester Stallone]] (as part of the ''Rocky'' dolls line, Jakks Pacific)

955 In 1991, Earl founded [[Planet Hollywood]],&lt;ref name=JC59364/&gt; negotiating celebrity investments from [[Arnold Schwarzenegger]] and [[Bruce Willis]], and partnerships{{clarification¦date=April 2023}} including [[Demi Moore]], and [[Sylvester Stallone]]. The company has filed for [[Chapter 11]] bankruptcy twice, but Earl still owns a large amount of stock with original investors Schwarzenegger and Willis, alongside new Asian investors.&lt;ref name=BizWk/&gt;

956 * [[Sylvester Stallone]]

957 '''''The Specialist''''' is a 1994 American [[action film¦action]] [[thriller film]] directed by [[Luis Llosa]] and starring [[Sylvester Stallone]], [[Sharon Stone]], [[James Woods]], [[Eric Roberts]], and [[Rod Steiger]].&lt;ref&gt;{{cite news¦title= Call in the Specialist Series: On the Job. Spotlighting the Workaday World. One in a Series.¦work= [[The Los Angeles Times]]¦date=1994-06-18¦url= https://www.latimes.com/archives/la-xpm-1994-06-18-ca-5621-story.html ¦access-date=2010-12-25¦first=Mike¦last=Clary}}&lt;/ref&gt; It is loosely based on &quot;The Specialist&quot; series of novels by [[John Shirley]]. The film was met with negative critical response, but became a box office success, and [[Gloria Estefan]]'s version of &quot;[[Turn the Beat Around#Gloria Estefan version¦ Turn the Beat Around]]&quot; became a dance sensation, becoming #1 on ''[[Billboard chart¦Billboard]]'''s [[Dance Club Songs]] chart.

958 * [[Sylvester Stallone]] as [[Captain (armed forces)¦Captain]] Ray Quick aka [[Specialist (rank)¦The Specialist]]

959 [[Steven Seagal]] was offered to star in and direct the film. Seagal was a fan of the script, but he wanted the sex scenes toned down, and wanted $9 million. The studio didn't want to pay Seagal that much money, and offered it to [[Sylvester Stallone]] instead.{{Citation needed¦date=October 2023}}

960 * Worst Actor ([[Sylvester Stallone]]) - Nominated

961 * Worst Actor (Sylvester Stallone) - Nominated

962 According to an interview with actor Loren Avedon,&lt;ref&gt;{{cite web¦url=http://cityonfire.com/feature-loren-avedon-interview//¦title=Interview with Loren Avedon at cityonfire.com}}&lt;/ref&gt; he shot his scenes independently with his scenes involving Richard Jaeckel as his boss. In addition, in the scene where Avedon's character does the [[Sylvester Stallone¦Stallone]]-like scream after seeing Khan and remembering the death of his brother, Avedon told director Lo he wanted to just toss the film tape into the fire in anger. He also did not get along well with actress Sherrie Rose. Avedon claimed Rose was acting like a &quot;prima donna&quot; and told her to &quot;shut up&quot; and that she had bad-mouthed him afterwards.

963 Ralph is the first composer in the history of the [[Academy Awards¦Oscars]] to have received multiple nominations for Best Original song that originated from documentaries.&lt;ref&gt;{{cite web¦ last1=Anita¦first1=Busch¦title=Oscar Nom Reactions: Saoirse Ronan, Leonardo DiCaprio, Bryan Cranston, Sylvester Stallone &amp; More¦url=https://deadline.com/2016/01/oscar-nominees-reactions-2016-academy-award-nominations-1201682531/¦website=deadline¦date=14 January 2016 ¦publisher=deadline.com¦access-date=15 January 2016}}&lt;/ref&gt; Additionally, Ralph is the first composer in [[Academy Awards¦Academy Award's]] history to write and produce the original music for multiple Oscar-nominated documentary films in the same year and the first composer to have back to back nominations for best song in a documentary.&lt;ref&gt;{{cite magazine¦last1=Neman¦first1=Melinda¦title=J. Ralph &amp; Antony Hegarty's Oscar-Nominated Song Was Inspired By Singing of Extinct Bird¦url=https://www.billboard.com/articles/news/6843700/j-ralph-antony-hegarty-oscar-nominated-manta-ray-racing-extinction¦magazine=Billboard¦access-date=January 14, 2016}}&lt;/ref&gt;

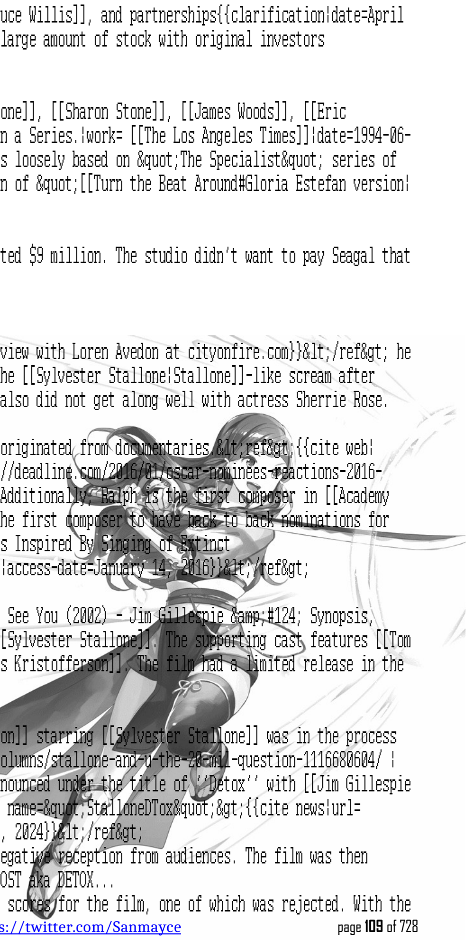
964 * [[Sylvester Stallone]]

965 '''''D-Tox''''' is a 2002 American [[thriller film]]&lt;ref&gt;{{cite web ¦ url=https://www.allmovie.com/movie/eye-see-you-v181893 ¦ title=Eye See You (2002) - Jim Gillespie &amp;#124; Synopsis, Movie Info, Moods, Themes and Related &amp;#124; AllMovie }}&lt;/ref&gt; directed by [[Jim Gillespie (director)¦Jim Gillespie]] and starring [[Sylvester Stallone]]. The supporting cast features [[Tom Berenger]], [[Charles S. Dutton]], [[Polly Walker]], [[Robert Patrick]], [[Stephen Lang]], [[Jeffrey Wright]], [[Courtney B. Vance]] and [[Kris Kristofferson]]. The film had a limited release in the United States on September 20, 2002, under the title '''''Eye See You''''' by [[DEJ Productions]].

966 * [[Sylvester Stallone]] as Agent Jake Malloy

967 In December 1997, it was announced a [[Brian Grazer]] produced film from [[Imagine Entertainment]] about [[Witness protection¦Witness Protection]] starring [[Sylvester Stallone]] was in the process of being greenlit at [[Universal Pictures¦Universal]].&lt;ref name=&quot;StalloneU&quot;&gt;{{cite news¦url= https://variety.com/1997/voices/columns/stallone-and-u-the-20-mil-question-1116680604/ ¦ title= Stallone and U: the $20 mil question ¦publisher=Variety¦access-date=July 19, 2024}}&lt;/ref&gt; In July 1998, the film was formally announced under the title of ''Detox'' with [[Jim Gillespie (director)¦Jim Gillespie]] slated to direct from a script by Ron Brinkenhoff who'd previously worked as a staff researcher for Imagine.&lt;ref name=&quot;StalloneDTox&quot;&gt;{{cite news¦url= https://variety.com/1998/film/news/stallone-enters-detox-1117478708/ ¦title=   Stallone enters 'Detox' ¦publisher=Variety¦access-date=July 19, 2024}}&lt;/ref&gt;

968 After the film was finished in 1999, Universal decided to screen it to a test audience but all the screenings of the first cut were met with negative reception from audiences. The film was then shelved for quite some time while re-shoots and story changes were being done.&lt;ref&gt;{{cite web¦title=Universal's EYE SEE YOU aka THE OUTPOST aka DETOX... info¦url=http://www.aintitcool.com/node/8011¦website=Aint It Cool¦accessdate=24 May 2017}}&lt;/ref&gt; Composer John Powell wrote two complete scores for the film, one of which was rejected. With the



film delayed and relegated to a European release by Universal due to the studio's dissatisfaction with the film in general, most of Powell's score was replaced with additional music by William Ross, Geoff Zanelli, and Nick Glennie-Smith as an attempt to make the film salvageable. A new ending was also filmed in which the main villain is killed in a different way.&lt;ref&gt;{{cite web|last1=Fischer|first1=Paul|title=Sylvester Stallone for &quot;D-Tox&quot;|url=http://www.darkhorizons.com/sylvester-stallone-for-d-tox/|website=Dark Horizons|date=6 September 2002|accessdate=24 May 2017}}&lt;/ref&gt; Even after re-shoots and title changes, Universal did not care for the film and after test screenings for the new version also got negative response from audiences, they shelved it. DEJ Productions acquired domestic distribution rights from Universal and released it over three years after it was originally finished but in a limited release.&lt;ref&gt;{{cite web|last1=Goldstein|first1=Patrick|title=Hidden by Hollywood: All-star cast of bombs|url=https://www.latimes.com/archives/la-xpm-2002-oct-15-et-goldside15-story.html|website=Los Angeles Times|date=15 October 2002 |access-date=24 May 2017}}&lt;/ref&gt;

In an interview with ''[[Ain't It Cool News]]'' in December 2006, Sylvester Stallone was asked why the film did not get a wide release and then answered:

During the original filming of ''D-Tox'' in 1999, Sylvester Stallone became attached to star in another Universal produced film, an action horror entitled ''Fatalis''. Written by novelist [[Jeff Rovin]] in 1998 and sold to Universal for over a million dollars, the script for the film concerned a huge pack of [[Smilodon|saber-toothed tigers]] who come back to life after an [[El Niño]] hurricane awakens them from being frozen inside an ancient sinkhole for thousands of years. They start attacking any humans they run into while moving down the mountains and through the woods before eventually attacking [[Los Angeles]].&lt;ref&gt;{{Cite web|url=https://gointothestory.blcklst.com/the-definitive-spec-script-sales-list-1991-2012-1998-918113f4199a|title = The Definitive Spec Script Sales List (1991-2012): 1998|date = 10 April 2018}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.empireonline.com/movies/news/eye-tiger/|title=Eye of the Tiger|date=22 October 1998 }}&lt;/ref&gt; Rovin, who wrote the novelization of Stallone's ''[[Cliffhanger (film)|Cliffhanger]]'' (1993) and his biography, wrote ''Fatalis'' with Stallone in mind for the lead role of an anthropologist who, along with a female reporter he befriends, wants to capture the tigers alive as they are our last link with the past, however the local sheriff wants to destroy them. The film went into pre-production by 2000, but following the huge failure of ''D-Tox'' and Stallone's other films, Universal ceased production and it went unproduced, though Rovin later turned his script into very successful novel of the same name. Rovin did the same with ''Vespers'', another cancelled action-horror film from around the same time, which would have focused on [[Pteropus|giant bats]] that attack [[New York City]]. When asked in 2021 about the chances of ''Fatalis'' being produced twenty years later, Rovin revealed that Stallone still owns the rights to the original story, and he suspects it will [[development hell|not be produced]].&lt;ref&gt;{{Cite web|url=https://stallonezone.com/fatalis061100.html|title = Stallone Zone News: Fatalis Now Available}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://stallonezone.com/wordpress/?page_id=32|title = &quot;Unmade&quot;|date = 11 October 2008}}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://www.goodreads.com/book/show/1081426.Fatalis | title=Fatalis }}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.goodreads.com/author/11762.Jeff_Rovin/questions|title = Jeff Rovin answers your questions - Ask the Author}}&lt;/ref&gt;

The film has an approval rating of 17% on [[Rotten Tomatoes]] based on 23 reviews, with an average rating of 3.5/10. The site's critical consensus reads, &quot;Whether it's being presented as ''D-Tox'' or ''Eye See You'', this Stallone starring vehicle is a slapdash thriller to actively avoid.&quot;&lt;ref&gt;{{cite web |title=D-Tox (Eye See You) |url=https://www.rottentomatoes.com/m/d_tox |website=Rotten Tomatoes |publisher=[[Fandango Media]] |access-date=10 November 2023}}&lt;/ref&gt; Danny Graydon of [[BBC Films]] said: &quot;Clearly, Hollywood's confidence in this film is lower than Pee-Wee Herman's Oscar chances, and their instincts are right: a boring, formulaic mix of serial killers and stalk'n'slash, this will not reinvigorate Sylvester Stallone's action hero status or loosen his maniacal destruction of the quality control button&quot;.&lt;ref&gt;{{cite news | first=Danny | last=Graydon | title=D-Tox (2002) | url=https://www.bbc.co.uk/films/2002/01/22/d-tox_review_2002_review.shtml | work=BBC Movies | publisher=British Broadcasting Corporation | date=22 January 2002 | accessdate=7 July 2012 }}&lt;/ref&gt;

| screenplay   = {{ubl|[[Phil Alden Robinson]]|[[Sylvester Stallone]]}}

* Sylvester Stallone

'''''Rhinestone''''' is a 1984 American [[musical film|musical]] [[comedy film]] directed by [[Bob Clark]] from a screenplay by [[Sylvester Stallone]] and [[Phil Alden Robinson]] and starring Stallone, [[Dolly Parton]], [[Richard Farnsworth]] and [[Ron Leibman]]. It is based on the 1975 hit song &quot;[[Rhinestone Cowboy]]&quot; written by [[Larry Weiss]]. Although a critical and financial failure, the film spawned two top 10 country hits for Parton.

* [[Sylvester Stallone]] as Nick Martinelli

[[File:Dolly Parton and Sylvester Stallone.jpg|thumb|upright|Parton and Stallone at a benefit showing of the film]]

Sylvester Stallone was tentatively set to do a film for Paramount before beginning production on ''Rhinestone'' for Fox. The Paramount producers, [[Don Simpson]] and [[Jerry Bruckheimer]], were encountering the same problems as their Fox counterparts, after offering Stallone the lead in ''[[Beverly Hills Cop]],'' Stallone told the Paramount executives he wanted to make a few changes and then proceeded to rewrite the script. Simpson and Bruckheimer, with more experience producing big-budget, high concept projects, than the executives at Fox, responded by saying that they didn't want to film the script as rewritten and promptly exited talks with Stallone and his representatives.&lt;ref name=&quot;:0&quot;&gt;{{Cite book |last=Litwak |first=Mark |url=https://archive.org/details/reelpowerstruggl00litw |title=Reel Power: The Struggle for Influence and Success in the New Hollywood |publisher=William Morrow and Company |year=1986 |isbn=9780688048891 |location=New York City, New York |pages=28 |language=English}}&lt;/ref&gt;

According to Robinson, Stallone did two drafts, one worse than the other. Stallone had, in a few short weeks, thrown out what had taken Robinson a year to write. Stallone changed the attitude of his character, making him more streetwise and less vulnerable.&lt;blockquote&gt;''&quot;Stallone's script read very much like a first draft'',&quot; according to one observer. “''You didn't sense a lot of work went into the choices. There had been a lot of texture in the screenplay, a lot of charm, and the characters were very well defined and they went through carefully plotted-out progressions of character change. All that was gone''.” The original script worked hard to keep out clichés. Stallone added them.&lt;/blockquote&gt;Fox executives agreed that Stallone's changes by and large hurt the script. With seven weeks to go before the start of principal photography, the studio said they would not make the movie on the basis of the revised script. Phil Alden Robinson was brought back in and asked to ''“restore the charm''” of the original, and “''put those great characters back''',” without changing any of the dialogue Sylvester Stallone had written for himself. Robinson objected, telling Fox executive Joe Wizan that what was wrong with the script was the awful dialogue. Wizan replied: “''Welcome to Hollywood''.” Robinson reluctantly agreed to the studio's restrictions on the rewrite. He says he was able to solve some structural problems and tried to make sense of Stallone's changes. When he turned it in, the studio said the new script was great, and one executive vowed that Stallone was either going to do it as written or walk. When they showed the changes to Stallone he said, “''Fine, fine. I just have a few changes to make''.” Stallone then proceeded to restore most of his changes.

Three weeks into production, Don Zimmerman was fired. His contract was settled and the production shut down while the studio sought a new director. [[Creative Artists Agency|CAA]] proposed that Fox hire [[Bob Clark]], who had directed the hit film Porky's, which Fox had distributed&lt;ref&gt;{{Cite book |last=Litwak |first=Mark |title=Reel Power: The Struggle for Influence and Success in the New Hollywood |publisher=William Morrow and Company |year=1986 |isbn=9780688048891 |location=New York City, New York |pages=25-26 |language=English}}&lt;/ref&gt;-contrary to Stallone's later statement's, director [[Mike Nichols]] was never considered for, much less affiliated with ''Rhinestone''. Nichols was still very loyal to his agent [[Sam Cohn]], with [[ICM Partners|ICM]], throughout the 1980s;&lt;ref&gt;{{Cite news |last=Weinraub |first=Bernie |date=August 23, 1993 |title=The Talk of Hollywood; Movers and Shakers Shaken Up By a Move |pages=11 |work=The New York Times |url=https://www.nytimes.com/1993/08/23/movies/the-talk-of-hollywood-movers-and-shakers-shaken-up-by-a-move.html |access-date=July 3, 2023 |archive-url=https://archive.today/20230629005202/https://www.nytimes.com/1993/08/23/movies/the-talk-of-hollywood-movers-and-shakers-shaken-up-by-a-move.html |archive-date=June 29, 2023}}&lt;/ref&gt; and CAA oversaw principle talent (package) acquisitions on ''Rhinestone'', during this time. Since the new director had to be acceptable to Sylvester Stallone, it didn't hurt that both were represented by CAA. Clark reportedly received $1 million to direct, plus a guaranteed fee for his next project at Fox, whether or not it was ever produced. That project is reputed to have been previously rejected by the studio. The first three weeks of filming were thrown out (all footage directed by Zimmerman, including location work in New York City and Nashville). Knowing his reputation for problem sets, Sylvester Stallone was anxious not to be cast as the villain in this production switch so he told a reporter that the change had been made because studio executives were panicked when the film fell behind schedule.&lt;ref&gt;{{Cite book |last=Daly |first=Marsha |url=https://archive.org/details/sylvesterstallon0000daly |title=Sylvester Stallone |publisher=St. Martin's Press |year=1986 |isbn=9780312903589 |edition=mass market |location=New York City, New York |pages=134 |language=English}}&lt;/ref&gt;

Phil Alden Robinson publicly distanced himself from the film during its release.&lt;ref name=&quot;rhine&quot;/&gt; Robinson took the highly unusual step of mounting his own publicity campaign, criticizing the movie, saying that the humor and intelligence of the original script had been replaced with vulgarity, caricature and farce. Disassociating himself from the picture, he sent letters to film critics explaining what had happened to his script. It was, he said, a textbook example of a studio willingly sacrificing the quality of a script for what was perceived as the marketability of star casting.&lt;ref name=&quot;:0&quot; /&gt;&lt;blockquote&gt;&quot;''I thought a long time before speaking out'',&quot; Robinson says. &quot;''Of course the conventional wisdom is that you don't open your mouth. But I was having a hard time living with that. I wrote letters and I gave interviews and I did a lot of press. I couldn't do nothing. I mean, I was doing nothing and I couldn't keep doing that. I would rather take the heat for saying what I believe and for telling the truth than to just let the industry know you can do this to me. And so I did a lot of press and got no heat. Stallone called my agent and was furious but that was it. People at studios, even at Fox, called to say we're glad you did that and it's about time somebody said these things. Writers, of course, were thrilled. I actually got calls from people who worked on the movie, from agents, and who were allied with Stallone, saying, we're glad you did that, we can't say this to Sly, but you're right.''&quot; &lt;ref name=&quot;:1&quot; /&gt;&lt;/blockquote&gt;In her 1994 autobiography, Parton said of the film “I guess the public didn't want to see Sylvester Stallone do comedy - or see me do Sylvester Stallone.&quot;&lt;ref&gt;{{cite magazine |first=Stephen |last=Betts |title=The 50 Best Dolly Parton Songs|url=https://www.rollingstone.com/music/music-lists/best-dolly-parton-songs-1234701711/tennessee-homesick-blues-1234701802/ |magazine=Rolling Stone |date = April 5, 2023|access-date=April 14, 2023}}&lt;/ref&gt;

| [[Sylvester Stallone]]

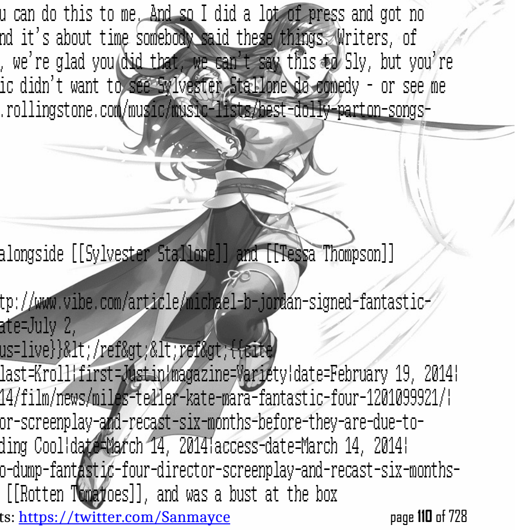
| Screenplay by [[Phil Alden Robinson]] and Sylvester Stallone; &lt;br&gt; Story by Phil Alden Robinson

{{Sylvester Stallone}}

[[Category:Films with screenplays by Sylvester Stallone]]

[[File:Sylvester Stallone, Tessa Thompson, and Michael B. Jordan promoting Creed at the Philadelphia Art Museum.JPG|thumb|right|Jordan alongside [[Sylvester Stallone]] and [[Tessa Thompson]] promoting ''[[Creed (film)|Creed]]'' in November 2015]]

In 2015, he starred as Johnny Storm, the Human Torch, in ''[[Fantastic Four (2015 film)|Fantastic Four]]''.&lt;ref&gt;{{cite web|url=http://www.vibe.com/article/michael-b-jordan-signed-fantastic-four|title=Michael B. Jordan Signed On For 'Fantastic Four'|publisher=vibe|date=October 21, 2013|access-date=February 3, 2014|archive-date=July 2, 2015|archive-url=https://web.archive.org/web/20150702142908/http://www.vibe.com/article/michael-b-jordan-signed-fantastic-four|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine|url=https://variety.com/2014/film/news/miles-teller-kate-mara-fantastic-four-1201099921/|title='Fantastic Four' Cast Revealed|last=Kroll|first=Justin|magazine=Variety|date=February 19, 2014|access-date=February 20, 2014|archive-date=August 3, 2020|archive-url=https://web.archive.org/web/20200803065433/https://variety.com/2014/film/news/miles-teller-kate-mara-fantastic-four-1201099921/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.bleedingcool.com/2014/03/14/fox-looking-to-dump-fantastic-four-director-screenplay-and-recast-six-months-before-they-are-due-to-shoot/|title=Fox Looking To Dump Fantastic Four Director, Screenplay And Recast, Six Months Before They Are Due To Shoot|publisher=Bleeding Cool|date=March 14, 2014|access-date=March 14, 2014|archive-date=March 15, 2014|archive-url=https://web.archive.org/web/20140315103807/http://www.bleedingcool.com/2014/03/14/fox-looking-to-dump-fantastic-four-director-screenplay-and-recast-six-months-before-they-are-due-to-shoot/|url-status=live}}&lt;/ref&gt; The film was universally panned by critics, holding a 9% approval rating on [[Rotten Tomatoes]], and was a bust at the box



office.&lt;ref&gt;{{Citation|title=Fantastic Four|url=https://www.rottentomatoes.com/m/fantastic_four_2015/|access-date=January 14, 2017|archive-date=September 1, 2015|archive-url=https://web.archive.org/web/20150901022151/http://www.rottentomatoes.com/m/fantastic_four_2015|url-status=live}}&lt;/ref&gt; However, later in 2015, Jordan rebounded with critical acclaim when he starred as [[Donnie Creed]], the son of boxer [[Apollo Creed]] in the seventh ''[[Rocky (film series)|Rocky]]'' film, ''[[Creed (film)|Creed]]'', his second collaboration with Coogler, which co-starred [[Sylvester Stallone]].&lt;ref&gt;{{cite web | title=Creed Movie - Michael B. Jordan and Sylvester Stallones - Spoiler Alert | work=phillychitchat.com | url=http://www.phillychitchat.com/2015/02/breaking-news-re-creed-movie-michael-b.html | access-date=February 24, 2015 | archive-url=https://web.archive.org/web/20150924072352/http://www.phillychitchat.com/2015/02/breaking-news-re-creed-movie-michael-b.html | archive-date=September 24, 2015 | url-status=dead }}&lt;/ref&gt; Jordan prepared for his role as a boxer in ''Creed'' by undertaking one year of rigorous physical training and a stringent low-fat diet.&lt;ref name=&quot;:0&quot;&gt;{{Cite news|last=Buckley|first=Cara|date=October 28, 2015|title=Michael B. Jordan Gives Millennials Their 'Rocky' With 'Creed'|newspaper=[[The New York Times]]|url=https://www.nytimes.com/2015/11/01/movies/michael-b-jordan-gives-millennials-their-rocky-with-creed.html|url-access=subscription|access-date=October 29, 2015|issn=0362-4331|archive-date=October 29, 2015|archive-url=https://web.archive.org/web/20151029151100/http://www.nytimes.com/2015/11/01/movies/michael-b-jordan-gives-millennials-their-rocky-with-creed.html|url-status=live}}&lt;/ref&gt; He did not have a body double during filming and was &quot;routinely bloodied, bruised, and dizzy&quot; when fighting scenes were being filmed.&lt;ref name=&quot;:0&quot; /&gt;

987 Devin dated rock star [[Prince (musician)|Prince]] in 1985 and [[Sylvester Stallone]] in 1988. She married model Randall Sanford in 1989, then divorced him before she married actor and musician [[Ronn Moss]] on September 25, 2009.&lt;ref name=&quot;canyon&quot;&gt;{{cite web|title=Tina Hillstrom Throws Shower For Ronn Moss's Bride To Be|url=http://www.canyon-news.com/artman2/publish/On_the_Industry_1168/Tina_Hillstrom_Throws_Shower_For_Ronn_Moss_s_Bride_To_Be.php|url-status=dead|archiveurl=https://web.archive.org/web/20110708115257/http://www.canyon-news.com/artman2/publish/On_the_Industry_1168/Tina_Hillstrom_Throws_Shower_For_Ronn_Moss_s_Bride_To_Be.php|archivedate=July 8, 2011|accessdate=February 3, 2010}}&lt;/ref&gt;

988 He appeared mostly in low-budget action and [[martial arts]] films such as ''Jungle Rats'' (1987), and ''Black Fire'' (1985), during the 80's, often directed by [[Teddy Page]] and [[John Gale (director)]]. The vast majority of Kristoff's films were produced by [[K.Y. Lim]]'s ''Silver Star Film Company'' (also called ''Kinavesa'' in the Philippines), a prolific, if somewhat notorious producer of Filipino [[action films]]. The ''Silver Star'' films usually recycled the same cast of [[United States|American]] and [[Europe]]an [[Z-movie]] workhorses from one film to another. Some of the recurring faces were [[Mike Monty]], James Gaines, [[Mike Cohen (actor)|Mike Cohen]], Gwendolyn Hung,&lt;ref&gt;{{Cite web|title = Gwendolyn Hung|url = https://www.imdb.com/name/nm0177059/|website = IMDb|access-date = 2016-02-14}}&lt;/ref&gt; [[Frank Juhas]] and [[Ronnie Patterson]], and Kristoff acted with all of them. He became something of a star for ''Silver Star'', appearing in several leading roles throughout the 80's, most prominently in ''Slash'' (1984), a film influenced by the ''[[Rambo (film series)|Rambo]]'' films of [[Sylvester Stallone]] to the point of plagiarism.

989 The idea of revamping the waterfront came from cities like [[Baltimore, Maryland|Baltimore]], who were always trying to fix formerly horrible places.&lt;ref name=&quot;Weinstein&quot;/&gt; [[Chris Turner (author)|Chris Turner]], the author of the book ''[[Planet Simpson]]'', writes about the scene at the Squidport: &quot;The 'satirical' setting seems almost documentary&quot;. The Squidport is a local revitalization project, which recast a historic industrial area as a pedestrian mall. He calls this an example of how &quot;hyper-consumer culture of Springfield moves front and center&quot;.{{sfn|Turner|2004|p=106}} On the waterfront, Rainier Wolfcastle opens a restaurant called &quot;Planet Hype&quot;.{{sfn|Turner|2004|p=106}} This is a parody of the international theme restaurant franchise [[Planet Hollywood]].&lt;ref name=&quot;Weinstein&quot; /&gt; Wolfcastle is a parody of [[Arnold Schwarzenegger]],{{sfn|Turner|2004|p=385}} who launched Planet Hollywood along with [[Sylvester Stallone]], [[Bruce Willis]], [[Demi Moore]] and [[Whoopi Goldberg]] in 1991.&lt;ref&gt;{{cite news|url=https://slate.com/business/2003/08/arnold-s-campaign-the-new-planet-hollywood.html|title=Arnold's Bad Business|author=Gross, Daniel|work=Slate|date=August 11, 2003|access-date=January 22, 2022|archive-date=January 22, 2022|archive-url=https://web.archive.org/web/20220122173525/https://slate.com/business/2003/08/arnold-s-campaign-the-new-planet-hollywood.html|url-status=live}}&lt;/ref&gt; According to [[Matt Groening]], the show had written an entire episode around Planet Hollywood, which featured the voices of Schwarzenegger, Stallone, and Willis as [[The Three Stooges]] type of characters. The episode was never animated, because it turned out that it was only the publicist of Planet Hollywood's idea and the actors did not want to participate.&lt;ref&gt;{{cite AV media|last=Groening|first=Matt|date=2006|title=Commentary for &quot;My Sister, My Sitter&quot;. The Simpsons: The Complete Eighth Season|medium=DVD|publisher=20th Century Fox}}&lt;/ref&gt;

990 * Aleta Ogord appears in ''[[Guardians of the Galaxy Vol. 2]]'', portrayed by [[Michelle Yeoh]]. This version is a member of the [[Ravagers (Marvel Cinematic Universe)|Ravagers]]. In a mid-credits scene, she interacts with members of [[Yondu Udonta (Marvel Cinematic Universe)|Yondu Udonta]]'s original team.&lt;ref name=&quot;OldTeam&quot;&gt;{{cite web|url=http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu|title=Sylvester Stallone's 'Guardians of the Galaxy Vol. 2' character has a big future in the MCU|last=Daniell|first=Mark|work=The Whig|date=April 25, 2017|access-date=April 28, 2017|archive-url=https://web.archive.org/web/20170428051803/http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu|archive-date=April 28, 2017|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;RTCameos&quot;&gt;{{cite web|url=http://www.radiotimes.com/news/2017-04-30/the-11-best-easter-eggs-in-guardians-of-the-galaxy-vol-2 |title=The 11 best Easter eggs in Guardians of the Galaxy Vol. 2 |last=Fullerton |first=Huw |work=[[Radio Times]] |date=April 30, 2017 |access-date=May 5, 2017 |archive-url=https://web.archive.org/web/20170504101448/http://www.radiotimes.com/news/2017-04-30/the-11-best-easter-eggs-in-guardians-of-the-galaxy-vol-2 |archive-date=May 4, 2017 |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://movieweb.com/adam-warlock-guardians-galaxy-marvel-cinematic-universe-mcu/|title=Adam Warlock &amp; Original Guardians Join Marvel Cinematic Universe|date=20 April 2017|access-date=15 May 2017}}&lt;/ref&gt;

991 He has worked with directors, [[Ron Shelton]] (twice), [[Clint Eastwood]], [[Sylvester Stallone]], [[Bill L. Norton]] (five times), and [[Michael Mann (director)|Michael Mann]] (five times); one of his notable Mann roles was in ''[[Public Enemies (2009 film)|Public Enemies]]'', starring [[Johnny Depp]] as [[John Dillinger]]. Mann handpicked Bentt to play Herbert Youngblood, who, along with Dillinger, staged the infamous Crown Point Jail break. He also starred as Biggis (El Plaga) opposite [[Beanie Sigel]], Noriega, and [[Damon Dash]] in the Dash-directed hip hop cult classic ''[[State Property 2]]''.

992 Jason Isaac Cutler was born in [[Worcester, Massachusetts]] on August 3, 1973. He grew up in nearby [[Sterling, Massachusetts|Sterling]] and attended [[Wachusett Regional High School]] in [[Holden, Massachusetts|Holden]]. He began working in his brother's concrete construction business, Cutler Bros. Concrete, when he was 11 years old.&lt;ref&gt;{{Cite web |last=Merritt |first=Greg |date=January 18, 2024 |title=How Jay Cutler Grew Fast: Interview, Full Workout &amp; Diet |url=https://thebarbell.com/jay-cutler-bodybuilding/ |website=The Barbell |access-date=January 19, 2024}}&lt;/ref&gt; His &quot;first heroes&quot; growing up were [[Jean-Claude Van Damme]] and [[Sylvester Stallone]].&lt;ref&gt;{{cite news |last1=Biswas |first1=Tiyasa |date=April 25, 2023 |title=&quot;My First Heroes&quot;: Not Arnold Schwarzenegger, but 12-Year-Old Jay Cutler Fell in Love With These Hollywood Superstars' Physique |url=https://www.essentiallysports.com/us-sports-news-bodybuilding-news-my-first-heroes-not-arnold-schwarzenegger-but-year-old-jay-cutler-fell-in-love-with-these-hollywood-superstars-physique |work=EssentiallySports |access-date=March 20, 2024}}&lt;/ref&gt; He started training to be a bodybuilder at the age of 18. He graduated from [[Quinsigamond Community College]] in 1993 with a degree in [[criminal justice]], intending to work as a corrections officer for a [[Prison#Security levels|maximum security prison]].

993 * [[Sylvester Stallone]]

994 '''''Judge Dredd''''' is a 1995 American [[science fiction film|science fiction]] [[action film]] based on the [[2000 AD (comics)|''2000 AD'' comics]] [[Judge Dredd|character of the same name]]. It is directed by [[Danny Cannon]] and stars [[Sylvester Stallone]] as the eponymous Judge Dredd, a law enforcement officer in the crime-ridden futuristic metropolis of [[Mega-City One]]. [[Armand Assante]], [[Diane Lane]], [[Rob Schneider]], [[Joan Chen]], [[Jürgen Prochnow]], and [[Max von Sydow]] appear in supporting roles. It was filmed entirely at [[Shepperton Studios]] in the United Kingdom, and released by [[Buena Vista Pictures]] on June 30, 1995.

995 Reviewers criticized the film for its script and perceived lack of originality and faith to its source material, along with Stallone's acting. The film is often considered to be one of Stallone's worst films,&lt;ref&gt;{{Cite web|url=http://www.flickchart.com/Charts.aspx?genre=Sci-Fi+Action&amp;perpage=100&amp;order=desc|title=The Top 100 Worst Sci-Fi Action Movies of All Time|website=Flickchart|access-date=2017-03-31}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |date=August 9, 2010 |author=Mike Thompson |url=http://www.metacritic.com/feature/sylvester-stallone-best-and-worst-movies|title=Sylvester Stallone: All Films Considered |work=[[Metacritic]] |access-date=2017-03-31 }}&lt;/ref&gt; but its visual style, effects, music score, stunts and action sequences were praised, and the film was nominated for four [[Saturn Award]]s. A reboot film, entitled ''[[Dredd]]'', was released in 2012.

996 * [[Sylvester Stallone]] as [[Judge Dredd|Judge Joseph Dredd]]

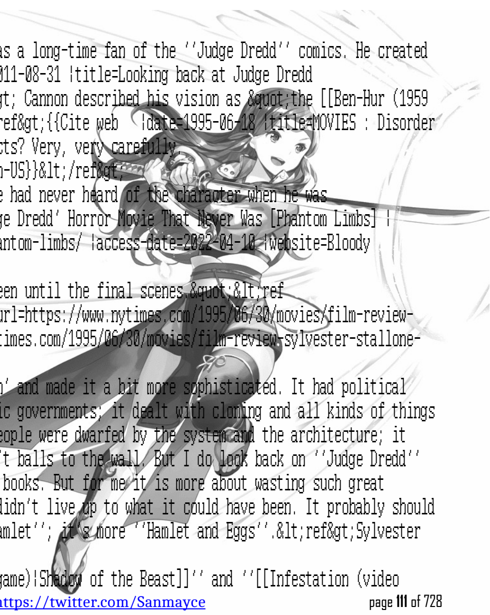
997 Director [[Danny Cannon]] was hired on the strength of his previous film, [[The Young Americans (film)|''The Young Americans'']]. Cannon was a long-time fan of the ''Judge Dredd'' comics. He created a mock-up poster for a ''Judge Dredd'' film that was published in Prog 534 of 2000 A.D. dated August 8, 1987.&lt;ref&gt;{{Cite web |date=2011-08-31 |title=Looking back at Judge Dredd |url=https://www.denofgeek.com/movies/looking-back-at-judge-dredd/ |access-date=2022-04-10 |website=Den of Geek |language=en-US}}&lt;/ref&gt; Cannon described his vision as &quot;the [[Ben-Hur (1959 film)|''Ben-Hur'']] of comic book movies,&quot; and turned down an offer to direct ''[[Die Hard with a Vengeance]]'' to make the film.&lt;ref&gt;{{Cite web |date=1995-06-18 |title=MOVIES : Disorder in the Court : On location : How does 'Judge Dredd's' crew of relative unknowns keep up with Sylvester Stallone and a slew of special effects? Very, very carefully |url=https://www.latimes.com/archives/la-xpm-1995-06-18-ca-14236-story.html |access-date=2022-04-10 |website=Los Angeles Times |language=en-US}}&lt;/ref&gt;

998 Early in development, [[Arnold Schwarzenegger]] was considered for title role, before [[Sylvester Stallone]] was cast, even though Stallone had never heard of the character when he was cast.&lt;ref&gt;{{Cite web |last=Jenkins |first=Jason |date=2020-07-14 |title=Writer Peter Briggs Remembers the Arnold Schwarzenegger 'Judge Dredd' Horror Movie That Never Was [Phantom Limbs] |url=https://bloody-disgusting.com/editorials/3623437/writer-peter-briggs-remembers-arnold-schwarzenegger-judge-dredd-horror-movie-never-phantom-limbs/ |access-date=2022-04-10 |website=Bloody Disgusting! |language=en-US}}&lt;/ref&gt; Cannon initially pursued [[Joe Pesci]] for the part of Fergie, but the actor turned it down.

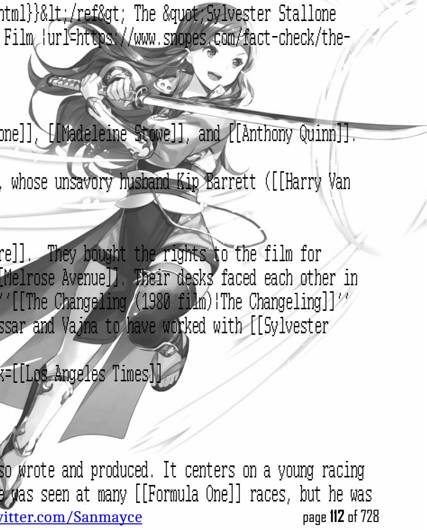
999 Caryn James of ''[[The New York Times]]'' wrote: &quot;Although it is full of noise and fake firepower, Dredd simply lies there on the screen until the final scenes.&quot;&lt;ref name=&quot;nytimes&quot;&gt;{{cite web |date=30 June 1995 |last=James |first=Caryn |title=FILM REVIEW; Sylvester Stallone, Judge. Uh-Oh. |url=https://www.nytimes.com/1995/06/30/movies/film-review-sylvester-stallone-judge-uh-oh.html |website=[[The New York Times]] |archive-url=https://web.archive.org/web/20091117192810/https://www.nytimes.com/1995/06/30/movies/film-review-sylvester-stallone-judge-uh-oh.html |archive-date=2009-11-17 |url-access=registration |url-status=live }}&lt;/ref&gt;

1000 &lt;blockquote&gt; I loved that property when I read it, because it took a genre that I love, what you could term the 'action morality film' and made it a bit more sophisticated. It had political overtones. It showed how if we don't curb the way we run our judicial system, the police may end up running our lives. It dealt with archaic governments; it dealt with cloning and all kinds of things that could happen in the future. It was also bigger than any film I've done in its physical stature and the way it was designed. All the people were dwarfed by the system and the architecture; it shows how insignificant human beings could be in the future. There's a lot of action in the movie and some great acting, too. It just wasn't balls to the wall. But I do look back on ''Judge Dredd'' as a real missed opportunity. It seemed that lots of fans had a problem with Dredd removing his helmet, because he never does in the comic books. But for me it is more about wasting such great potential there was in that idea; just think of all the opportunities there were to do interesting stuff with the Cursed Earth scenes. It didn't live up to what it could have been. It probably should have been much more comic, really humorous, and fun. What I learned out of that experience was that we shouldn't have tried to make it ''Hamlet''; it's more ''Hamlet and Eggs''.&lt;ref&gt;Sylvester Stallone interviewed in ''Uncut'' #131 (April 2008), p.118&lt;/ref&gt;&lt;/blockquote&gt;

1001 ''Nitro'' was also available as part of the Psygnosis monster pack, volume I, along with the original ''[[Shadow of the Beast (1989 video game)|Shadow of the Beast]]'' and ''[[Infestation (video



```
game)|Infestation]]''.  The &quot;monsters&quot; in the game may refer to the players-namely, the [[Man with No Name|Man with No Name (Clint Eastwood)]] (P1-white), [[John Rambo|Rambo (Sylvester Stallone)]] (P2-yellow), and [[James Bond|James Bond (Roger Moore)]] (P3-red).
1002 | caption = North American NES cover artwork featuring [[Sylvester Stallone]] as Ranger Gabriel &quot;Gabe&quot; Walker
1003 '''''Score''''' is a 1974 [[Erotic film|erotic]] [[romance film]] directed by [[Radley Metzger]].&lt;ref name=&quot;PB-201406&quot;&gt;{{cite web|last=Bentley|first=Toni|authorlink=Toni Bentley|title=The Legend of Henry Paris|url=http://www.playboy.com/articles/the-legend-of-henry-paris|date=June 2014|work=Playboy|accessdate=January 26, 2016|archive-date=August 1, 2016|archive-url=https://web.archive.org/web/20160801131922/http://www.playboy.com/articles/the-legend-of-henry-paris|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;TB-201406&quot;&gt;{{cite web|last=Bentley|first=Toni|authorlink=Toni Bentley|title=The Legend of Henry Paris|url=http://www.tonibentley.com/pdfarticles/playboy/RadleyMetzger_AuteuroftheErotic_ToniBentley.pdf|date=June 2014|website=ToniBentley.com|accessdate=January 26, 2016|archive-date=September 9, 2016|archive-url=https://web.archive.org/web/20160909125210/http://www.tonibentley.com/pdfarticles/playboy/RadleyMetzger_AuteuroftheErotic_ToniBentley.pdf|url-status=live}}&lt;/ref&gt; One of the first films to explore bisexual relationships,&lt;ref name=&quot;TRR-20170404&quot;&gt;{{cite web |author=Staff |title=Radley Metzger's 'Score' (1974): Behind the Scenes |url=http://www.therialtoreport.com/2017/04/04/score/ |date=April 4, 2017 |work=[[The Rialto Report]] |accessdate=April 4, 2017 |archive-date=January 12, 2024 |archive-url=https://web.archive.org/web/20240112063801/https://www.therialtoreport.com/2017/04/04/score/ |url-status=live }}&lt;/ref&gt;&lt;ref name=&quot;CINE-20140108&quot;&gt;{{cite web |last=Kramer |first=Gary M. |title=Interview: Radley Metzger, dir. of Score |url=http://cinedelphia.com/interview-radley-metzger-dir-of-score/ |date=January 8, 2014 |work=Cinedelphia |accessdate=April 4, 2017 |archive-date=October 17, 2021 |archive-url=https://web.archive.org/web/20211017070850/http://cinedelphia.com/interview-radley-metzger-dir-of-score/ |url-status=live }}&lt;/ref&gt; it was part of the brief [[porn chic]] fad of the [[Golden Age of Porn]] in the early 1970s that also included ''[[Behind the Green Door]]'', ''[[The Devil in Miss Jones]]'' and ''[[Deep Throat (film)|Deep Throat]]''. The film was based on an [[off-Broadway]] stage play that ran for 23 performances at the Martinique Theatre from October 28, 1970, through November 15, 1971 and featured [[Sylvester Stallone]] in a brief role (as telephone repairman Mike).&lt;ref name=&quot;FM-Summer1997&quot;&gt;{{citation|last=Gallagher|first=Steve|title=The Libertine|url=http://filmmakermagazine.com/archives/issues/summer1997/metzger.php|date=Summer 1997|work=[[Filmmaker (magazine)|Filmmaker Magazine]]|accessdate=May 24, 2015|archive-date=September 22, 2015|archive-url=https://web.archive.org/web/20150922002127/http://filmmakermagazine.com/archives/issues/summer1997/metzger.php|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;TRR-20170404&quot;/&gt;&lt;ref&gt;{{Cite news |last=Gussow |first=Mel |date=1970-10-29 |title=Theater: 'Score' Presented |language=en-US |work=The New York Times |url=https://www.nytimes.com/1970/10/29/archives/theater-score-presented-nude-sex-play-opens-at-the-martinique.html |access-date=2023-10-31 |issn=0362-4331 |archive-date=2023-10-31 |archive-url=https://web.archive.org/web/20231031114726/https://www.nytimes.com/1970/10/29/archives/theater-score-presented-nude-sex-play-opens-at-the-martinique.html |url-status=live }}&lt;/ref&gt; The theatrical version of ''Score'' was written by [[Jerry Douglas (director)|Jerry Douglas]], who later became a mainstream screenwriter. It was set in a shabby [[Queens]] tenement, while the film was set in an elegant, mythical land and sported a relatively high budget for an independent film of that era.
1004 [[Steve Rossi]] claimed he was offered the Lou Canova role but Allen reneged when Allen was told that the film would be known as an Allen and Rossi film.&lt;ref&gt;{{cite web|url=http://televisionario.onsugar.com/Interview-Steve-Rossi---Part-One-14922682|title=An Interview with Steve Rossi - Part One|author=BattleMN|work=onsugar.com|access-date=2011-08-10|archive-url=https://web.archive.org/web/20131215153216/http://televisionario.onsugar.com/Interview-Steve-Rossi---Part-One-14922682|archive-date=2013-12-15|url-status=dead}}&lt;/ref&gt; [[Robert De Niro]] and [[Sylvester Stallone]] both turned down the role.&lt;ref name=&quot;Mell2005&quot;&gt;{{cite book|author=Eila Mell|title=Casting Might-Have-Beens: A Film by Film Directory of Actors Considered for Roles Given to Others|url=https://books.google.com/books?id=8AMyBgAAQBAJ&amp;pg=PA45|date=6 January 2005|publisher=McFarland|isbn=978-0-7864-2017-9|pages=45-|access-date=14 October 2016|archive-date=21 September 2021|archive-url=https://web.archive.org/web/20210921110337/https://books.google.com/books?id=8AMyBgAAQBAJ&amp;pg=PA45|url-status=live}}&lt;/ref&gt;
1005 Casting started to take place, and Alexander Salkind wanted a famous actor in the role of [[Superman]]. [[Dustin Hoffman]] was once again highly considered though the idea was dropped. [[Robert Redford]] was offered the part, but he felt he was too famous and was not right for the role, as did [[Burt Reynolds]]. After the success of ''[[Rocky]]'', [[Sylvester Stallone]] lobbied hard for the role but was ignored. [[Paul Newman]] was offered all three roles of Superman, Jor-El and Lex Luthor to his choosing though he did not accept any of them, for a salary of $4 million.&lt;ref name=ilyandpierre/&gt; [[Nick Nolte]]&lt;ref name=flight&gt;{{cite video | title = Taking Flight: The Development of Superman | type = DVD | publisher = [[Warner Brothers]] |date = 2001}}&lt;/ref&gt; and [[Rex Harrison]] would both turn it down.&lt;ref name=ilyandpierre/&gt; It was decided to focus on casting [[Jor-El]] and [[Lex Luthor]] and save Superman for later. Hoffman was then offered the role of Luthor, and although he was interested, he turned it down. [[Marlon Brando]] would eventually be cast as Jor-El, causing [[Gene Hackman]] to be cast as Luthor as he wanted desperately to work with Brando. Impressed by the producers' ability to cast famous actors in respective roles, [[Warner Brothers|Warner Bros.]] decided to distribute the film internationally, rather than simply [[Domestic market|domestically]].&lt;ref name=cinematic/&gt;
1006 | Description       = Box cover of [[Sylvester Stallone]]'s first film, [[The Party at Kitty and Stud's|Party at Kitty and Stud's]].
1007 [[Category:Images of Sylvester Stallone]]</text>
1008 | starring       = [[Sylvester Stallone]]
1009 '''''The Party at Kitty and Stud's'''''{{hsp}}{{efn|name=fn1|Contemporary newspaper listings give the film's full title as ''The Party at Kitty and Stud's '''Place'''''.}} is a 1970 American [[softcore pornography|softcore]] [[Pornographic film|pornographic]] [[romance film]] directed, written &amp; produced by Morton Lewis and starring [[Sylvester Stallone]] in his first starring and leading role.  Stallone worked two days and was paid $200. Shortly after the 1976 release of ''[[Rocky]]'', ''The Party at Kitty and Stud's'' was edited and re-released as '''''Italian Stallion''''' to capitalize on its now-famous star.
1010 * [[Sylvester Stallone]] as Stud
1011 According to Stallone, the film was financed by &quot;a group of wealthy lawyers, very, very solid.&quot; The actor said in a 1978 ''[[Playboy]]'' interview that he had done the film out of desperation after being bounced out of his apartment and finding himself [[homelessness|homeless]] for several days, sleeping in a New York City bus station in the middle of winter. In Stallone's words: &quot;It was either do that movie or rob someone because I was at the end-at the very end-of my rope. Instead of doing something desperate, I worked two days for $200 and got myself out of the bus station&quot;. The actor also called the film &quot;horrendous&quot; and commented &quot;By today's standards, the movie would almost qualify for a [[PG-Rating#Ratings|PG rating]]&quot;.&lt;ref name=&quot;playboy&quot;&gt;{{cite magazine|author=&lt;!--Staff writer(s); no by-line.--&gt;|title=The Playboy interview: Sylvester Stallone|url=http://www.playboy.com/arts-entertainment/features/sylvester-stallone/sylvester-stallone-04.html|date=September 1978|magazine=[[Playboy]]|publisher=[[Playboy Enterprises]]|accessdate=9 November 2017|archiveurl=https://web.archive.org/web/20110424055949/http://www.playboy.com/arts-entertainment/features/sylvester-stallone/sylvester-stallone-04.html|archivedate=24 April 2011}}&lt;/ref&gt;
1012 Although some sources, including Stallone himself,&lt;ref name=&quot;Ebert&quot;&gt;{{cite news |last1=Ebert |first1=Roger |authorlink1=Roger Ebert |title=Sylvester Stallone souring on stardom |url=https://www.newspapers.com/clip/23837115/ |work=Arizona Daily Star |date=June 18, 1977 |page=C6 |via=Newspapers.com |access-date=October 28, 2018 |archive-date=August 31, 2024 |archive-url=https://web.archive.org/web/20240831154743/https://www.newspapers.com/article/arizona-daily-star/23837115/ |url-status=live }}&lt;/ref&gt; have stated that ''The Party at Kitty and Stud's''  was never released until after his success in ''Rocky'' (1976), contemporary newspaper advertisements indicate that the film was screened in [[Adult movie theater|adult cinemas]] in at least four U.S. states between 1970 and 1972.&lt;ref&gt;{{cite news |title=The Pottstown Mercury |url=https://www.newspapers.com/clip/23837040/ |date=September 23, 1970 |location=Pennsylvania |page=18 |via=Newspapers.com |access-date=October 28, 2018 |archive-date=August 31, 2024 |archive-url=https://web.archive.org/web/20240831154629/https://www.newspapers.com/article/the-mercury-party-at-kitty-and-studs-pl/23837040/ |url-status=live }}
1013 Stallone has stated that after ''Rocky'' became a hit, the owners of ''The Party at Kitty and Stud's'' offered to sell him the rights to the film for about US$100,000 in order to prevent its re-release, but the actor &quot;wouldn't buy it for two bucks&quot;.&lt;ref name=&quot;playboy&quot;/&gt; The film was then rereleased in theaters under the new title ''The Italian Stallion''. In a prologue included with the reissued version, and in the film's trailer, pornography director [[Gail Palmer]] sat by an [[Film editing|editing machine]] addressing the audience and presented the film as [[X-rated]].&lt;ref&gt;{{YouTube|5Md91CfO-lo|Promo for Italian Stallion}}.&lt;/ref&gt; The release campaign implied that the original film was [[hardcore pornography]], but that the material had now been edited into a milder version.&lt;ref&gt;{{Cite web |url=http://www.mondo-digital.com/sickpicks3.html |title=Mondo Digital's Notes on Digital Sleaze |access-date=2008-03-15 |archive-date=2012-08-01 |archive-url=https://web.archive.org/web/20120801051722/http://www.mondo-digital.com/sickpicks3.html |url-status=live }}&lt;/ref&gt; However, the truth of this version has been repeatedly challenged, notably by trade journal ''[[AVN (magazine)|AVN]]'', which examined an original print, finding no trace of hardcore scenes.&lt;ref name=AVNhoax1&gt;{{cite web|last=Sullivan|first=David|date=6 February 2008|title=The 'Italian Stallion' hoax: Stallone never did hardcore|website=AVN Business|publisher=[[Adult Video News]]|url=http://business.avn.com/articles/28745.html|accessdate=9 November 2017|archivedate=23 August 2009|archiveurl=https://web.archive.org/web/20090823060517/http://business.avn.com/articles/28745.html}}&lt;/ref&gt; The &quot;Sylvester Stallone porno movie&quot; evolved over the years into a minor [[urban legend]].&lt;ref name=&quot;Snopes&quot;&gt;{{cite web |title=Sylvester Stallone Porn Film |url=https://www.snopes.com/fact-check/the-italian-stallion/ |website=[[Snopes.com]] |accessdate=1 November 2018 |date=December 4, 2017 |archive-date=18 March 2018 |archive-url=https://web.archive.org/web/20180318002851/https://www.snopes.com/fact-check/the-italian-stallion/ |url-status=live }}&lt;/ref&gt;
1014 * [[Sylvester Stallone]]
1015 '''''Avenging Angelo''''' is a 2002 American [[direct-to-video]] [[action comedy film]] directed by [[Martyn Burke]] and starring [[Sylvester Stallone]], [[Madeleine Stowe]], and [[Anthony Quinn]]. The film received mostly negative reviews.
1016 Frankie Delano ([[Sylvester Stallone]]) has spent his life safeguarding Angelo as well as Angelo's daughter, Jennifer Barrett ([[Madeleine Stowe]]), whose unsavory husband Kip Barrett ([[Harry Van Gorkum]]) has had their young son Rawley (Ezra Perlman) placed in a boarding school against Jennifer's wishes.
1017 * [[Sylvester Stallone]] as Frankie Delano
1018 The first film Kassar and Vajna ventured together was ''[[The Sicilian Cross]]'', a 1976 [[Italian language|Italian]] film that starred [[Roger Moore]].  They bought the rights to the film for $130,000. Kassar flew to [[Asia]] and sold it for $220,000.&lt;ref name=ew/&gt; By the early 1980s, Vajna and Kassar had bought a small office in [[Melrose Avenue]]. Their desks faced each other in the office and Vajna's wife and Kassar's girlfriend were their secretaries.&lt;ref name=ew/&gt;  Kassar and Vajna served as executive producers on ''[[The Changeling (1980 film)|The Changeling]]'' (1980), ''[[The Amateur (1981 film)|The Amateur]]'' (1981), and ''[[Escape to Victory]]'' (1981). The latter film marked the first time for both Kassar and Vajna to have worked with [[Sylvester Stallone]].&lt;ref name=dofg/&gt;
1019 | screenplay     = [[Sylvester Stallone]]&lt;ref&gt;{{cite news |date=September 20, 2001 |author=SUSAN KING |title=''Driven'' Starts Up on DVD |work=[[Los Angeles Times]] |url=https://www.latimes.com/archives/la-xpm-2001-sep-20-ca-47597-story.html |access-date=20 May 2021 }}&lt;/ref&gt;
1020 * Sylvester Stallone
1021 * Sylvester Stallone
1022 '''''Driven''''' is a 2001 American [[Action film|action]] [[sports film]] directed by [[Renny Harlin]] and starring [[Sylvester Stallone]], who also wrote and produced. It centers on a young racing driver's effort to win the [[Championship Auto Racing Teams|CART FedEx Championship Series]] auto racing championship. Prior to production, Stallone was seen at many [[Formula One]] races, but he was
```



unable to procure enough information about the category due to the secrecy with which teams protect their cars, so he decided to base the film on [[Championship Auto Racing Teams|CART]].

1023 * [[Sylvester Stallone]] as Joe &quot;The Hummer&quot; Tanto, a veteran Champ Car and Indy 500 champion who comes out of retirement to train Jimmy

1024 &lt;blockquote&gt;And of those, about 20 were about this one man's journey, myself, through this film, and all his trials and tribulations. He'd fallen from a great height career-wise. He was a drunkard with all these problems and accidents because he and his wife Cathy, who's played by Gina Gershon, had this very tumultuous relationship. (Laughs) I'm giving you a little biographical hint here. And he just started to come apart. So he was brought back as kind of like how people should never be. It's like taking kids who are truants and then taking them to prison to see where they'll end up and scaring them straight. So I was brought back to basically prove to young Jimmy Bly how he should never be, as a bad example. And then the more we worked on it, it became the dark side, a little seedy, and I didn't know where the upside of it was ever going to be. So we began to reduce his role and make it more of an ensemble, so he's just there as a guy who did his job, wasn't very spectacular, would race like hell, sometimes he'd win, sometimes lose, but he had a certain work ethic code, that old school that could be applied to Jimmy. So that all made it more ensemble, and then in the editing we reduced it even more. I originally had a relationship going with the reporter. But that began to de-emphasise the other people, so we put that on the back burner... So we did shoot it but it we said, &quot;Nah, it's not really flying.&quot;&lt;ref name=&quot;cinema&quot;&gt;{{cite web|url=http://cinema.com/articles/340/driven-interview-with-sylvester-stallone.phtml|title=Driven : Interview With Sylvester Stallone}}&lt;/ref&gt;&lt;/blockquote&gt;

1025 Stallone and Harlin had previously worked together on ''Cliffhanger''. Harlin had been trying to develop a film on the life of [[Ayrton Senna]] and when that fell through signed on to make ''Driven''.&lt;ref&gt;{{cite web|url=https://news.google.com/newspapers?nid=1346&amp;dat=20010503&amp;id=OtROAAAAIBAJ&amp;sjid=__0DAAAAIBAJ&amp;pg=6241,946219&amp;hl=en|title=Lakeland Ledger - Google News Archive Search}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://au.ign.com/articles/2001/04/13/ign-filmforce-interview-with-sylvester-stallone|title=IGN FilmForce Interview with Sylvester Stallone|first=Steve|last=Head|date=13 April 2001 }}&lt;/ref&gt;

1026 When Sylvester Stallone's character arrives at a race early in the film, there's a long shot of him walking in, saying hello to various people. One very brief uncredited cameo contains Dustin Hoffman, in racing gear.

1027 {{Sylvester Stallone}}

1028 [[Category:Films with screenplays by Sylvester Stallone]]

1029 | rowspan=2 | [[Sylvester Stallone]]

1030 | [[Renny Harlin]], [[Elie Samaha]], [[Sylvester Stallone]]

1031 {{About|the singer|the Olympian|Mike Sylvester|the actor|Sylvester Stallone}}

1032 |'''[[Sylvester Stallone]]'''

1033 | rowspan=&quot;2&quot; |'''[[Sylvester Stallone]]'''

1034 | [[Sylvester Stallone]]

1035 | [[Sylvester Stallone]]

1036 |'''[[Sylvester Stallone]]'''

1037 | rowspan=&quot;2&quot; | [[Sylvester Stallone]]

1038 | [[Sylvester Stallone]]

1039 | [[Sylvester Stallone]]

1040 |'''[[Sylvester Stallone]]'''

1041 | [[Sylvester Stallone]]

1042 | rowspan=&quot;2&quot; | [[Sylvester Stallone]]

1043 | [[Sylvester Stallone]]

1044 | [[Sylvester Stallone]]

1045 | rowspan=&quot;3&quot; | [[Sylvester Stallone]]

1046 | [[Sylvester Stallone]]

1047 |[[Sylvester Stallone]]

1048 [[File:Sylvester Stallone Comic-Con 2010.jpg|thumb|150px|right|[[Sylvester Stallone]] holds the records for most total nominations (16), consecutive nominations (9, 1984-92), and wins (4).]]

1049 *[[Sylvester Stallone]]

1050 *[[Sylvester Stallone]]

1051 The gun, designed as a [[stakeout]] gun for easy hiding in civilian clothing, soon gained a criminal reputation, when a batch of 22 pieces was stolen from the workshop in 1984. The manufacturing license of the shop, Tampereen Asepaja, was revoked, and no pieces are produced anymore. The firearm can be seen in the 1984 movie ''[[Red Dawn]]'', as well as in the 1986 film ''[[Cobra (1986 film)|Cobra]]'', wielded by [[Sylvester Stallone]].&lt;ref&gt;{{Cite web|url=http://guns.connect.fi/gow/kysvast39.html|title = Gunwritersin Tokko tietoa on - osa 39}}&lt;/ref&gt; It is also used as the primary weapon of [[DC Comic]]s [[vigilante]] ''[[Wild Dog (comics)|Wild Dog]]''.

1052 {{Non-free video cover|image has rationale=yes|1980s American film video covers}}[[Category:Images of Sylvester Stallone]]</text>

1053 | starring   = [[Sylvester Stallone]] &lt;!--Only Stallone's name appears on the poster block. Do not add anyone else--&gt;

1054 '''''Oscar''''' is a 1991 American [[comedy film]] directed by [[John Landis]].&lt;ref name=&quot;Oscar&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/20385/Oscar/|title=Oscar|work=[[Turner Classic Movies]]|access-date=March 23, 2016}}&lt;/ref&gt; Based on the Claude Magnier stage play, it is a remake of the [[Oscar (1967 film)|1967 French film of the same name]], but set in [[Great Depression|Depression-era]] New York City. ''Oscar'' stars [[Sylvester Stallone]], in a rare comedic role, as Angelo &quot;Snaps&quot; Provolone, a [[mob boss]] who promises his dying father that he will leave the world of crime and become an honest businessman. Alongside Stallone, the film's cast includes [[Marisa Tomei]], [[Ornella Muti]], [[Tim Curry]] and [[Chazz Palminteri]].&lt;ref name=&quot;Kehr&quot;&gt;{{cite web|url=https://www.chicagotribune.com/1991/04/26/rehashing-oscar-is-no-snap-for-stallone-and-landis/|title=Rehashing 'Oscar' Is No Snap For Stallone And Landis|last=Kehr|first=Dave|author-link=Dave Kehr|date=April 26, 1991|website=[[Chicago Tribune]]|access-date=January 3, 2013}}&lt;/ref&gt; Its score was composed by [[Elmer Bernstein]].

1055 * [[Sylvester Stallone]] as Angelo &quot;Snaps&quot; Provolone

1056 Conversely, ''Tribune'' reviewer [[Gene Siskel]] gave the film a score of three out of four stars. While he described the first reel as &quot;disastrous,&quot; he added that the film included &quot;truly funny work by enormously talented supporting players.&quot;&lt;ref name=&quot;Siskel&quot;&gt;{{cite news|title= Stallone's 'Oscar' Recovers From Bad Start|work= Chicago Tribune|date=April 26, 1991|first=Gene|last=Siskel|url= https://www.chicagotribune.com/1991/04/26/stallones-oscar-recovers-from-bad-start/|access-date=2013-01-03}}&lt;/ref&gt; [[Roger Ebert]] was in full agreement with Siskel on ''[[At the Movies (1986 TV program)|Siskel &amp; Ebert &amp; the Movies]]'' and they gave the film &quot;two thumbs up&quot;.&lt;ref name=&quot;Ebert&quot;&gt;{{cite web | url=https://siskelebert.org/?p=5831 |title=Oscar, Toy Soldiers, A Kiss Before Dying, A Rage in Harlem, 1991 |access-date=2021-02-16 |work=Siskel and Ebert Movie Reviews}}&lt;/ref&gt; ''[[Variety (magazine)|Variety]]'' stated the film was an &quot;intermittently amusing throwback to gangster comedies of the 1930s. While dominated by star Sylvester Stallone and heavy doses of production and costume design, pic is most distinguished by sterling turns by superb character actors.&quot;&lt;ref&gt;{{cite web|url=https://www.variety.com/review/VE1117793769/?refCatId=31|title=Review: 'Oscar'|date=December 31, 1990|website=[[Variety (magazine)|Variety]]|access-date=January 3, 2013|archive-url=https://web.archive.org/web/20140416114357/http://variety.com/1990/film/reviews/oscar-1200428799/|archive-date=April 16, 2014}}&lt;/ref&gt; Kathleen Maher of ''[[The Austin Chronicle]]'' gave the film three out of five stars, commending Stallone's performance: &quot;I'm not used to having much good to say about the guy, but Stallone has evidenced a nascent sense of humor before, and here he allows it to blossom.&quot;&lt;ref&gt;{{cite web|url=https://www.austinchronicle.com/events/film/1991-05-10/oscar/|title=Oscar|last=Maher|first=Kathleen|date=May 10, 1991|website=[[The Austin Chronicle]]|access-date=March 7, 2021}}&lt;/ref&gt;

1057 ''Oscar'' was nominated for three [[Razzie Awards]] at the [[12th Golden Raspberry Awards]] in 1992: Worst Actor (Sylvester Stallone), Worst Director (John Landis) and Worst Supporting Actress (Marisa Tomei).&lt;ref name=razzies1992-03-29/&gt;

1058 :{{cquote|I made a movie once, ''Oscar'' with Sylvester Stallone, and everybody who saw the movie and liked it, would never go out and see Stallone. We did a preview of the movie and someone wrote on the card, 'Why didn't he take his shirt off and kill anybody?' [Laughs.] But we had an extraordinary cast, we had Kirk Douglas, Don Ameche, and a girl who had never been in a movie before: Marisa Tomei. She was so extremely great from the first day! And what had she done before? Practically nothing! Now, that's talent. She was so unbelievably good.&lt;ref&gt;{{cite web|url=https://filmtalk.org/2017/09/06/john-landis-im-always-taking-great-pride-in-the-fact-that-i-hope-i-dont-have-a-style-of-my-own/|title=John Landis: &quot;I'm always taking great pride in the fact that-I hope-I don't have a style of my own&quot;|date=September 6, 2017|website=FilmTalk.org|access-date=February 24, 2019}}&lt;/ref&gt;}}

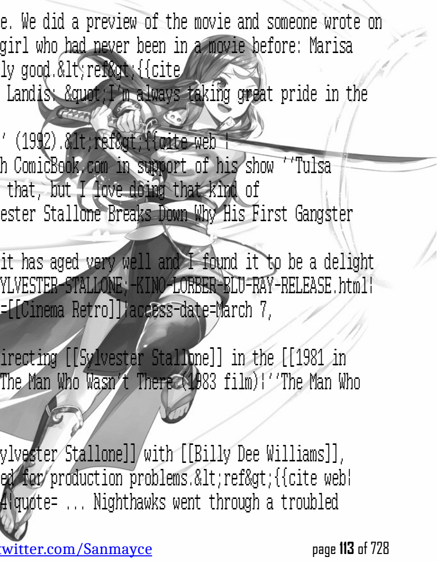
1059 In 2006, Stallone was asked about what films he wished he wouldn't have done and mentioned ''Oscar'' right after ''[[Stop! Or My Mom Will Shoot]]'' (1992).&lt;ref&gt;{{cite web | url=https://legacy.aintitcool.com/node/30899 | title=Round #5 - Stallone keeps slugging out answers to the AICN Mob!!! }}&lt;/ref&gt; Speaking with ComicBook.com in support of his show ''Tulsa King'', Stallone says he still loves ''Oscar'' in spite of its lukewarm reception: "I guess it was too much of a shocking transition from Rambo to that, but I love doing that kind of drama."&lt;ref&gt;{{cite web | url=https://comicbook.com/movies/news/tulsa-king-sylvester-stallone-why-oscar-flopped/ | title=Tulsa King Star Sylvester Stallone Breaks Down Why His First Gangster Movie Flopped }}&lt;/ref&gt;

1060 In 2020, Lee Pfeiffer of ''[[Cinema Retro]]'' wrote that &quot;''Oscar'' was a box office flop and critics attacked it across the board. However, it has aged very well and I found it to be a delight throughout. Give it chance, will ya?&quot;&lt;ref&gt;{{cite web|url=https://cinemaretro.com/index.php?/archives/10926-REVIEW-OSCAR-1991-STARRING-SYLVESTER-STALLONE;-KINO-LORBER-BLU-RAY-RELEASE.html|title=Review: &quot;Oscar&quot; (1991) Starring Sylvester Stallone; Kino Lorber Blu-ray Release|last=Pfeiffer|first=Lee|date=July 24, 2020|website=[[Cinema Retro]]|access-date=March 7, 2021}}&lt;/ref&gt;

1061 Malmuth's directoral debut was ''[[Fore Play]]'', an anthology comedy film which he co-directed with [[John G. Avildsen]]. He was best known for directing [[Sylvester Stallone]] in the [[1981 in film|1981]] thriller ''[[Nighthawks (1981 film)|Nighthawks]]'' and [[Steven Seagal]] in the 1990 political action film ''[[Hard to Kill]]'' and [[The Man Who Wasn't There (1983 film)|''The Man Who Wasn't There'']], starring [[Steve Guttenberg]].

1062 * [[Sylvester Stallone]]

1063 '''''Nighthawks''''' is a 1981 American [[neo-noir]] [[action film|action]] [[crime thriller film]] directed by [[Bruce Malmuth]] and starring [[Sylvester Stallone]] with [[Billy Dee Williams]], [[Lindsay Wagner]], [[Persis Khambatta]], [[Nigel Davenport]], and [[Rutger Hauer]]. Its score was composed by [[Keith Emerson]]. The film was noted for production problems.&lt;ref&gt;{{cite web|title=Top 10 Sylvester Stallone Movies|date=10 July 2014|url=http://www.top10hq.com/top-10-sylvester-stallone-movies/|access-date=17 September 2014|quote= ... Nighthawks went through a troubled production cycle filled with rewrites and reshoots.}}&lt;/ref&gt;



1064 * [[Sylvester Stallone]] as Sergeant Deke DaSilva

1065 The original director was [[Gary Nelson (director)¦Gary Nelson]]-who had directed the [[Disney]] films ''[[Freaky Friday (1976 film)¦Freaky Friday]]'' (1976) and ''[[The Black Hole]]'' (1979)-but he was dismissed from the project after a week of production and was not credited. His replacement, [[Bruce Malmuth]],&lt;ref&gt;{{cite news¦title= Bruce Malmuth, 71; Directed Thrillers and Documentaries, Acted in 'The Karate Kid' ¦work= [[The Los Angeles Times]]¦date=July 3, 2005¦url= https://www.latimes.com/archives/la-xpm-2005-jul-03-me-malmuth3-story.html¦access-date=2010-11-25}}&lt;/ref&gt; had only one previous film to his credit: a segment of the 1975 [[portmanteau]] comedy, ''[[Fore Play]]''. Malmuth, en route from Los Angeles to New York City, was unable to make the first day of shooting after Nelson's removal and Stallone shot a scene for one day (the chase in the subway). According to the [[Directors Guild of America]], &quot;Anyone signed to work on a movie before the director was engaged cannot replace a fired director, except in an emergency&quot;; arbitration resulted in a fine.&lt;ref name=&quot;afi.com&quot;/&gt; In preparation for their roles as New York City police officers, Billy Dee Williams and Sylvester Stallone spent several weeks working at night with the New York [[Street Crimes Unit]].

1066 &lt;blockquote&gt;The stunts in the film were pretty extraordinary because they were invented along the way. Running through the tunnels of an un-built subway station was very dangerous, but exciting and we were only given one hour to do it. So that made for an interesting evening. Hanging from the cable car was probably one of the more dangerous stunts I was asked to perform because it was untested and I was asked to hold a folding [[Gerber Legendary Blades¦Gerber knife]] in my left hand so if the cable were to snap, and I survived the 230 foot fall into the [[East River]] with its ice cold 8 mile an hour current, I could cut myself free from the harness because the cable when stretched out weighed more than 300 lbs. I tell you this because it's so stupid to believe that I would survive hitting the water so to go beyond that is absurd. So I actually thought the smart move would be to commit hari-kari on the way down and let the cards fold as they may. P.S. Several years later this cable did snap while testing it on a 100lb bag of sand.&lt;ref name= &quot;aicn&quot;&gt;{{cite news¦title=Round One With Sylvester Stallone Q&amp;A‼¦publisher=[[Ain't It Cool News]]¦ date=December 1, 2006¦ url=http://www.aintitcool.com/node/30861¦access-date=2008-06-19}}&lt;/ref&gt;&lt;/blockquote&gt;

1067 In a 1980 interview with [[Roger Ebert]], Stallone mentioned problems with stunts he wanted to perform himself in ''Nighthawks''. In the scene where he jumps onto a moving train and kicks out the wire-reinforced window, the window broke easily and knocked him off balance. Stallone has had a lifelong fear of heights and said about the helicopter stunt, &quot;I've never been so scared in my life&quot;. According to the actor, he spent 15 weeks in near-total seclusion in his hotel room between scenes and it was the most stressful time of his life.&lt;ref&gt;{{cite web¦url=http://www.rogerebert.com/interviews/interview-with-sylvester-stallone-1980¦title=Interview with Sylvester Stallone (1980) - Interviews - Roger Ebert¦first=Roger¦last=Ebert}}&lt;/ref&gt;

1068 In 2021 interview for Collider while he was promoting his new director's cut of ''[[Rocky IV]]'' (1985), Stallone said how some of his other films he would have loved to re-edit back to their original versions were ''[[Rocky V]]'' (1990), ''[[Paradise Alley]]'' (1978), and ''Nighthawks''. He also praised the original cut of ''Nighthawks'' as superior to the final theatrical version.&lt;ref&gt;{{Cite web ¦ url=https://www.youtube.com/watch?v=Gc6Txstbqgg ¦ title=Sylvester Stallone on His New Cut of Rocky 4 and the Other Films He'd Love to Re-Edit ¦ website=[[YouTube]] ¦ date=13 November 2021 }}&lt;/ref&gt;

1069 In her review for ''[[The New York Times]]'', [[Janet Maslin]] praised Hauer's performance: &quot;Mr. Hauer's terrorist, in particular, is a sharply drawn character who acts as a driving force within the movie's scheme. Sadism and bloodlessness are his only identifiable characteristics, and yet he behaves memorably wherever he goes&quot;.&lt;ref name= &quot;maslin&quot;&gt;{{cite news ¦ last = Maslin ¦ first = Janet ¦ title = ''Nighthawks'' ¦ work = [[New York Times]] ¦ date = April 10, 1981 ¦ url = https://movies.nytimes.com/movie/review?res=9C03E5DF153BF933A25757C0A967948260 ¦ access-date = 2008-06-19 }}&lt;/ref&gt; ''[[Time (magazine)¦Time]]'' magazine's [[Richard Schickel]] sharply criticized the film: &quot;''Nighthawks'' is so moronically written and directed, so entirely without wit or novelty, that there is plenty of time to wonder about its many missing explanations&quot;.&lt;ref name= &quot;schickel&quot;&gt;{{cite magazine ¦ last = Schickel ¦ first = Richard ¦ title = Chicken Feed ¦ magazine = [[Time (magazine)¦Time]] ¦ date = May 11, 1981 ¦ url = http://www.time.com/time/magazine/article/0,9171,949158,00.html ¦ archive-url = https://web.archive.org/web/20101015153617/http://www.time.com/time/magazine/article/0,9171,949158,00.html ¦ url-status = dead ¦ archive-date = October 15, 2010 ¦ access-date = 2009-12-17 }}&lt;/ref&gt; In ''[[The Globe and Mail]]'', [[Jay Scott]] wrote that the film &quot;has a dirty job to do and does it. That is not an endorsement. [[Thumbscrew (torture)¦Thumbscrews]] and [[cattle prods]] are real good at what they do, too&quot;.&lt;ref name= &quot;scott&quot;&gt;{{cite news ¦ last = Scott ¦ first = Jay ¦ title = The old it-takes-guts-to-kill story ¦ work = [[The Globe and Mail]] ¦ date = April 11, 1981 }}&lt;/ref&gt; According to ''[[Newsweek]]''{{'s}} Jack Kroll, &quot;This is one of those films that isn't a film but some repulsively complicated business deal&quot;.&lt;ref name= &quot;kroll&quot;&gt;{{cite news ¦ last = Kroll ¦ first = Jack ¦ title = Goose Chase ¦ work = [[Newsweek]] ¦ pages = 93 ¦ date = April 20, 1981 }}&lt;/ref&gt; In ''[[The Washington Post]]'', Gary Arnold described the film as &quot;an aggressively shallow police thriller pitting New York undercover cops against international terrorists, suggests what ''[[The Day of the Jackal (film)¦The Day of the Jackal]]'' might have looked like if filmed by the producers of ''[[Baretta]]''. In order to facilitate a grandstanding, harebrained heroic role assigned to Sylvester Stallone, the filmmakers brush off every opportunity for intelligent dramatization and authentic suspense that the plot would seem to possess&quot;.&lt;ref name= &quot;arnold&quot;&gt;{{cite news ¦ last = Arnold ¦ first = Gary ¦ title = ''Nighthawks'' Nosedives ¦ newspaper = [[Washington Post]] ¦ pages = C3 ¦ date = April 13, 1981 }}&lt;/ref&gt;

1070 In July 2019, it was announced that a remake was in development from [[Balboa Productions]].&lt;ref&gt;{{Cite web¦url=https://variety.com/2019/film/features/balboa-prods-sylvester-stallone-braden-aftergood-1203275756/¦title=Sylvester Stallone's Production Company Wants to be the Blumhouse of Action Films¦date=23 July 2019}}&lt;/ref&gt; By May 2020, Sylvester Stallone announced that a reboot television series is currently in development. The project will be a joint-venture production between [[Universal Television]] and Balboa Productions, and will be released as a [[Peacock (streaming service)¦Peacock]] exclusive [[Television show¦television series]].&lt;ref&gt;{{Cite web¦url=https://comicbook.com/movies/news/nighthawks-movie-tv-series-remake-universal-peacock/¦title = Sylvester Stallone Announces Nighthawks Remake as Streaming Series¦ date=4 May 2020 }}&lt;/ref&gt;

1071 Summerbee also starred in the cult film ''[[Escape to Victory]]'' alongside [[Sylvester Stallone]], [[Michael Caine]] and [[Pelé]].&lt;ref&gt;{{cite web ¦url=http://www.manchestereveningnews.co.uk/sport/football/football-news/escape-to-victory-film-remake-7338800 ¦title=Escape To Victory, again! ¦work=[[Manchester Evening News]] ¦date=27 June 2014 ¦ access-date=12 September 2015 }}&lt;/ref&gt;&lt;ref name=&quot;Small Talk&quot;&gt;{{cite web ¦url=https://www.theguardian.com/sport/2008/sep/05/smalltalk.mikesummerbee ¦title=Small Talk: Mike Summerbee ¦work=[[The Guardian]] ¦date=5 September 2008 ¦access-date=14 September 2018 }}&lt;/ref&gt;

1072 A broker was hired to market Metropolitan Tower's apartments; at the time, it was an unusual move for a condominium development in Manhattan, as most condominiums marketed their units with in-house staff.&lt;ref&gt;{{Cite news ¦last=Catalano ¦first=Joe ¦date=August 27, 1988 ¦title=Opting Early for Sales Help ¦pages=72 ¦work=Newsday ¦via=newspapers.com ¦url=https://www.newspapers.com/clip/83948803/opting-early-for-sales-help/ ¦access-date=August 22, 2021 ¦archive-date=August 22, 2021 ¦archive-url=https://web.archive.org/web/20210822222324/https://www.newspapers.com/clip/83948803/opting-early-for-sales-help/ ¦url-status=live }}&lt;/ref&gt; Just before the launch of sales, in late 1985, Macklowe began receiving offers for condominiums from &quot;celebrities, producers, and directors&quot;.&lt;ref name=&quot;p154403626&quot; /&gt; Among those potential tenants was actor [[Sylvester Stallone]].&lt;ref&gt;{{Cite news ¦last1=Kubasik ¦first1=Ben ¦last2=Scaduto ¦first2=Anthony ¦date=December 5, 1985 ¦title=Inside New York ¦pages=6 ¦work=Newsday ¦via=newspapers.com ¦ url=https://www.newspapers.com/clip/83928869/inside-new-york/ ¦access-date=August 22, 2021 ¦archive-date=August 22, 2021 ¦archive-url=https://web.archive.org/web/20210822174808/https://www.newspapers.com/clip/83928869/inside-new-york/ ¦url-status=live }}&lt;/ref&gt; The interest from the entertainment industry was in part because the building was near the offices of several entertainment companies including [[CBS]] and [[American Broadcasting Company¦ABC]].&lt;ref name=&quot;p154403626&quot; /&gt; By early 1986, Macklowe had sold 96 residential units.&lt;ref name=&quot;nyt19860320&quot; /&gt; Metropolitan Tower had been nicknamed &quot;the Russian Tea Room Annex&quot;, and a marketing manager said that about two-thirds of residential condominium buyers were Americans.&lt;ref name=&quot;International 1986&quot; /&gt; Construction was finished in 1987. That year, computer company [[Control Data Corporation]] leased {{cvt¦117000¦ft2}}, nearly half of the building's office space.&lt;ref&gt;{{Cite news ¦last=Barsky ¦first=Neil ¦date=March 10, 1987 ¦title=Bear Stearns' lair going Park Ave. ¦ pages=131 ¦work=New York Daily News ¦via=newspapers.com ¦url=https://www.newspapers.com/clip/83944844/bear-stearns-lair-going-park-ave/ ¦access-date=August 22, 2021 ¦archive-date=August 22, 2021 ¦ archive-url=https://web.archive.org/web/20210822211810/https://www.newspapers.com/clip/83944844/bear-stearns-lair-going-park-ave/ ¦url-status=live }}&lt;/ref&gt; One of the largest residential transactions at Metropolitan Tower occurred shortly after opening, when a single Japanese buyer purchased 36 apartments.&lt;ref&gt;{{Cite news ¦last=Lueck ¦first=Thomas J. ¦date=December 18, 1988 ¦ title=The Condo Glut That Didn't Happen ¦language=en-US ¦work=The New York Times ¦url=https://www.nytimes.com/1988/12/18/realestate/the-condo-glut-that-didn-t-happen.html ¦access-date=August 21, 2021 ¦issn=0362-4331 ¦archive-date=August 21, 2021 ¦archive-url=https://web.archive.org/web/20210821232434/https://www.nytimes.com/1988/12/18/realestate/the-condo-glut-that-didn-t-happen.html ¦url-status=live }}&lt;/ref&gt;

1073 :::*&quot;Han Solo (born 29 BBY), a character in the fictional Star Wars universe, is played by actor Harrison Ford&quot;, &quot;A film based around the comic strip was released in 1995, starring Sylvester Stallone as Dredd.&quot; -- Both quotes from Wikipedia articles on these characters. [[User:Anetode¦Anetode]] 23:33, 22 September 2005 (UTC)

1074 * A 1970 low-budget film starring Sylvester Stallone in his first film role, originally titled ''[[The Party at Kitty and Stud's]]''

1075 * [[Sylvester Stallone]] (born 1946), American actor, who played the character Rocky Balboa

1076 The neighborhood was the fictional setting of the movie [[Rocky]], by [[Sylvester Stallone]]. It is alternatively known as &quot;K&amp;A&quot; (for the intersection of Kensington and Allegheny Avenues.)

1077 Scott Weinberg of [[UGO Networks¦eFilmCritic.com]] gave the film 2 stars out of 5, writing that &quot;Norris does [[Sylvester Stallone¦Stallone]]... badly&quot; in his review.&lt;ref name=&quot;Rotten Tomatoes&quot;&gt;{{cite web¦url=https://www.rottentomatoes.com/m/1014008_missing_in_action ¦title= ''Missing in Action'' Movie Reviews, Pictures ¦publisher=[[Rotten Tomatoes]] ¦ access-date=December 21, 2019}}&lt;/ref&gt; In a 2003 [[BBC]] article entitled &quot;Rambo: Pretenders to the Throne&quot;, Almar Haflidason wrote that &quot;the runaway success of the ''Rambo'' trilogy inspired dozens of rip-offs&quot;, citing that the ''Missing in Action'' series was the most famous of the ''Rambo'' clones.&lt;ref&gt;{{cite web¦url=https://www.bbc.co.uk/films/2000/06/19/rambo_pretenders_article.shtml ¦title= Rambo: Pretenders to the Throne ¦publisher=BBC ¦access-date=July 2, 2010}}&lt;/ref&gt;

1078 * [[Armored (upcoming film)¦''Armored'' (upcoming film)]], film starring Sylvester Stallone

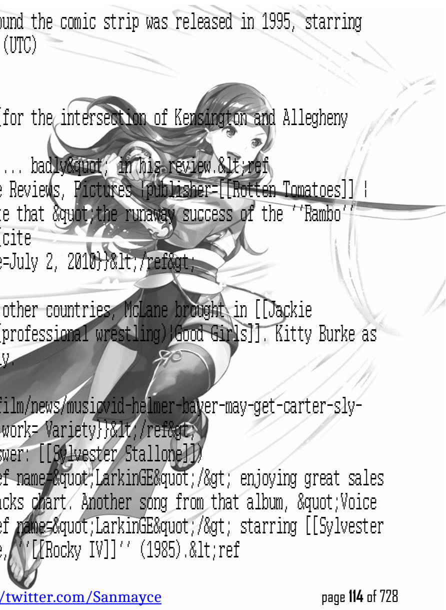
1079 The show was introduced at the 1986 [[NATPE]] Convention. Following the successful initial sale to 30 major television markets in the US and six other countries, McLane brought in [[Jackie Stallone]], mother of [[Sylvester Stallone]], to play [[kayfabe]] GLOW owner and the [[Manager (professional wrestling)¦manager]] of the [[Face (professional wrestling)¦Good Girls]]. Kitty Burke as Aunt Kitty, was the manager for the [[Heel (professional wrestling)¦Bad Girls]]. Stallone had been promoting a physical fitness gym for women only.

1080 * White Eagle Enterprises, a film production company founded by [[Sylvester Stallone]]

1081 *''[[Get Carter (2000 film)¦Get Carter]]'', the 2000 remake starring [[Sylvester Stallone]]&lt;ref&gt;{{cite web ¦url= https://variety.com/1999/film/news/musicvid-helmer-bayer-may-get-carter-sly-1117502718/¦title= Musicvid helmer Bayer may 'Get Carter' &amp; Sly¦date= June 3, 1999¦access-date= January 19, 2024¦first= Chris¦last= Petrikin¦work= Variety}}&lt;/ref&gt;

1082 * '''Question:''' Which major [[Planet Hollywood]] investor played the lead in the [[Get Carter (2000 film)¦2000 remake of ''Get Carter'']]? (Answer: [[Sylvester Stallone]])

1083 The group's follow-up album ''[[Tough All Over (John Cafferty &amp; The Beaver Brown Band album)¦Tough All Over]]'' (1985) made the top 40,&lt;ref name=&quot;LarkinGE&quot;/&gt; enjoying great sales for &quot;[[C-I-T-Y]]&quot; and the [[Tough All Over (song)¦title track]], which became their second number-one single on the Mainstream Rock Tracks chart. Another song from that album, &quot;Voice of America's Sons&quot;, was the featured theme song on the official soundtrack of the action movie ''[[Cobra (1986 film)¦Cobra]]'' (1986),&lt;ref name=&quot;LarkinGE&quot;/&gt; starring [[Sylvester Stallone]], and Cafferty's solo track &quot;[[Heart's on Fire (John Cafferty song)¦Heart's on Fire]]&quot; was featured in another Stallone movie, ''[[Rocky IV]]'' (1985).&lt;ref name=&quot;LarkinGE&quot;/&gt;



1084 Tarantino attempted to cast both [[Kal Penn]]&lt;ref&gt;{{cite news|last=Sciretta|first=Peter|url=https://www.slashfilm.com/article.php?story=20061121kalpenngrindhouse|archive-url=https://archive.today/20070722151341/http://www.slashfilm.com/article.php?story=20061121kalpenngrindhouse|url-status=dead|archive-date=July 22, 2007|title=Did You Know: Kal Penn was cast in Tarantino's ''Grindhouse''?|access-date=May 12, 2009|publisher=Slash Film}}&lt;/ref&gt; and [[Sylvester Stallone]]&lt;ref&gt;{{cite news|last=Sciretta|first=Peter|url=https://www.slashfilm.com/article.php?story=20061204stallonegrindhouse|archive-url=https://web.archive.org/web/20110511145304/http://www.slashfilm.com/article.php?story=20061204stallonegrindhouse|url-status=dead|archive-date=May 11, 2011|title=Tarantino wanted Stallone for ''Grindhouse''|access-date=May 12, 2009|publisher=Slash Film}}&lt;/ref&gt; in ''Death Proof'', but both were unable to work due to prior commitments. In an interview, Tarantino revealed that he decided to cast [[Kurt Russell]] as the killer stunt driver because &quot;for people of my generation, he's a true hero&amp;nbsp;... but now, there's a whole audience out there that doesn't know what Kurt Russell can do. When I open the newspaper and see an ad that says 'Kurt Russell in ''[[Dreamer (2005 film)|Dreamer]]'',' or 'Kurt Russell in ''[[Miracle (2004 film)|Miracle]]'',' I'm not disparaging these movies, but I'm thinking: When is Kurt Russell going to be a badass again?&quot;&lt;ref name=&quot;Bloodbath&quot;/&gt;

1085 Golan produced films featuring actors such as [[Sean Connery]], [[Sylvester Stallone]], [[Chuck Norris]], [[Jean-Claude Van Damme]], and [[Charles Bronson]], and for a period, was known as a producer of comic book-style films like ''[[Masters of the Universe (film)|Masters of the Universe]]'', ''[[Superman IV: The Quest for Peace]]'', ''[[Captain America (1990 film)|Captain America]]'', and his aborted attempt to bring [[Spider-Man]] to the silver screen. Golan also wrote and [[Screenwriting#polish|polished]] numerous film scripts under the pen name '''Joseph Goldman'''.&lt;ref&gt;Ronald Grover. [https://web.archive.org/web/20020421042136/http://www.businessweek.com/bwdaily/dnflash/apr2002/nf20020415_7441.htm &quot;Unraveling Spider-Man's Tangled Web&quot;. ''Business Week'' (April 15, 2002).] Retrieved on 2007-01-22.&lt;/ref&gt; At the time of his death, Golan had produced over 200 films, directed 44, and won 8 &quot;[[Kinor David]]&quot; awards as well as &quot;[[Israel Prize]]&quot; in Cinema. He was nominated for a [[BAFTA Award for Best Film Not in the English Language|BAFTA Award for Best Foreign-Language Film]] for [[Franco Zeffirelli]]'s ''[[Otello (1986 film)|Otello]]''.

1086 ''G.I. Joe: Order of Battle'' was a 4-issue mini series which ran from December 1986 - March 1987. Written by Larry Hama, with art by Herb Trimpe, the first issue spotlighted G.I. Joe characters with code names from A-K. Subsequent issues spotlighted G.I. Joe characters from M-Z, Cobra characters, and vehicles respectively. The second issue erroneously listed [[Sylvester Stallone]]'s [[Rocky Balboa]] character as a member of G.I. Joe. While negotiations had taken place to license the character, the deal had fallen through.&lt;ref&gt;{{cite web|url=http://www.yojoe.com/archive/unproduced/rockybalboa.shtml |title=Sculpt: Rocky Balboa |publisher=Yojoe.Com |access-date=2012-09-17}}&lt;/ref&gt;&lt;ref&gt;{{cite book |title= The Official 30th Anniversary Salute To G.I. Joe 1964-1994|last= Santelmo|first= Vincent|year= 1994|publisher= Krause Publications|isbn= 0-87341-301-6|page=112}}&lt;/ref&gt; The third and fourth issues contained a retraction stating that Rocky Balboa was not and had never been a member of G.I. Joe. A [[Trade paperback (comics)|trade-paperback]] including material from all four issues, was published in 1987, and removed mention of the Rocky character entirely.&lt;ref&gt;{{cite book|last=Hama|first=Larry|title=G.I. Joe: Order of Battle|publisher=[[Marvel Entertainment Group]]|year=1987|isbn=978-0-87135-288-0|author2=Herb Trimpe |author3=Jack Abel |display-authors=etal}}&lt;/ref&gt;

1087 WIBG and overnight talk show host Don Cannon were featured in the famous &quot;egg yolk drinking&quot; scene in the 1976 film ''[[Rocky]]'', playing in the background while the Philadelphia fighter (played by [[Sylvester Stallone]]) cracks six raw eggs into a glass and chugs them down.&lt;ref&gt;{{Cite web|url=http://www.phillymag.com/news/2014/08/22/don-cannon-philadelphia-radio-legend-passed/|title = Don Cannon, Philadelphia Radio Legend, Has Passed|date = 22 August 2014}}&lt;/ref&gt;

1088 The Gray Goose is also known as the equine star Sylvester Stallone, in the movie ''[[Sylvester (film)|Sylvester]]'', where he and Walnes doubled in the movie for the eventing shots. The movie included scenes from the pair's 1984 Rolex Three Day cross-country round.

```
1089       <text bytes="42970" sha1="otvhgqmhcgihzxsoad6b6abd0896y5g" xml:space="preserve">{{Short description|2006 film by Sylvester Stallone}}
1090 | director       = [[Sylvester Stallone]]
1091 | writer         = Sylvester Stallone
1092 | based_on       = {{Based on|[[List of Rocky characters|Characters]]|Sylvester Stallone}}
1093 * Sylvester Stallone
```

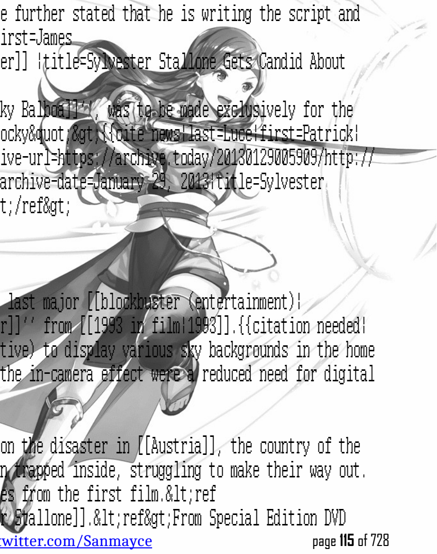
1094 '''''Rocky Balboa''''' &lt;!-- This film has never officially been called &quot;Rocky VI&quot;. --&gt; is a 2006 American [[Sports film|sports]] [[Drama (film and television)|drama film]] starring, written and directed by [[Sylvester Stallone]] in his first film as director [[Rocky IV|since 1985]]. It is the sequel to ''[[Rocky V]]'' (1990) and the sixth installment in the [[Rocky (franchise)|''Rocky'' film series]]. The film co-stars [[Burt Young]] and [[Antonio Tarver]] in his only acting role. In the film, [[Rocky Balboa]] (Stallone), now an aging small restaurant owner, is challenged to an [[exhibition fight]] by hothead young boxer Mason Dixon (Tarver).

1095 * [[Sylvester Stallone]] as [[Rocky Balboa|Robert &quot;Rocky&quot; Balboa]], retired boxer and former two-time heavyweight champion.

1096 {{quote|text=many athletes have a form of brain damage including football players, soccer players, and other individuals in contact sports such as rugby, etc. Rocky never went for a second opinion and yielded to his wife's wishes to stop. So with the advent of new research techniques into brain damage, Rocky was found to be normal among fighters, and he was suffering the results of a severe concussion. By today's standards Rocky Balboa would be given a clean bill of health for fighters.&lt;ref name=&quot;AICNQA1&quot;&gt;{{cite web|author=Moriaty|url=https://legacy.aintitcool.com/node/30861|title=Round One With Sylvester Stallone Q&amp;A!! |publisher=[[Ain't It Cool News]] |date=December 1, 2006|access-date=June 15, 2009}}&lt;/ref&gt;}}

1097 The film was an unexpected box office success and exceeded studio expectations grossing over three times the opening night estimates of (at best) $2,000,000 and doing so despite a harsh spell of winter weather.&lt;ref name=&quot;BOXSly&quot;&gt;{{cite news |date=December 21, 2006 |author=Harry Knowles |author-link=Harry Knowles |url=https://legacy.aintitcool.com/node/31054 |title=Sylvester Stallone public statement |publisher=[[Ain't It Cool News]] |access-date=June 15, 2009 }}&lt;/ref&gt; The film finished third in its opening weekend, grossing $12,540,000,&lt;ref name=&quot;BOXWeekend&quot;&gt;{{cite web|url=https://www.boxofficemojo.com/weekend/2006W51/|title=Weekend Box Office Results for December 22-24, 2007|website=[[Box Office Mojo]]|access-date=June 15, 2009}}&lt;/ref&gt;&lt;ref&gt;{{cite web |date=January 2, 2007 |author=Brandon Gray |title='Museum' Exhibits Holiday Clout |url=https://www.boxofficemojo.com/article/ed1417151492/?ref_=bo_ne_nl |website=[[Box Office Mojo]] |quote=nowhere near the league of the first four Rocky's, it's poised to sell more tickets than Rocky V.}}&lt;/ref&gt; and eventually became Stallone's most successful starring role since 1993's ''[[Cliffhanger (film)|Cliffhanger]]''&lt;ref&gt;{{cite web |url=https://www.boxofficemojo.com/name/nm0000230/ |title=Sylvester Stallone Movie Box Office Results |website=[[Box Office Mojo]] |access-date=June 15, 2009 }}&lt;/ref&gt; and the sixth highest grossing boxing film of all time, topped only by the first ''Rocky'' through ''Rocky IV'' and [[Clint Eastwood]]'s ''[[Million Dollar Baby]]''.&lt;ref name=&quot;BOXBoxing&quot;&gt;{{cite web |url=https://www.boxofficemojo.com/genres/chart/?id=boxing.htm |title=Sports - Boxing |website=[[Box Office Mojo]] |access-date=June 15, 2009 }} {{Dead link|date=August 2024}}&lt;/ref&gt; Total U.S. box office gross receipts were $70,269,899 while the international gross stands at $85,959,151 making for a total worldwide gross of $156,229,050.&lt;ref name=&quot;BOXTotal&quot;&gt;{{cite web |url=https://www.boxofficemojo.com/release/rl3077342721/ |title=Rocky Balboa ticket sales figures |website=[[Box Office Mojo]] |access-date=September 20, 2019 }}&lt;/ref&gt;

1098 In May 2019 at the [[Cannes Film Festival]], Sylvester Stallone said that he had another story about Rocky Balboa.&lt;ref&gt;{{cite web|url=https://www.msn.com/en-nz/news/other/sylvester-stallone-wants-to-make-a-new-and-really-different-rocky-movie-about-immigration/ar-AABVFOi#page=2|title=Sylvester Stallone wants to make a new and 'really different' Rocky movie about immigration|website=MSN|date=May 25, 2019|access-date=November 8, 2020}}&lt;/ref&gt; By July, Stallone confirmed that a sequel film series was in development. The project will be a joint-production venture between Winkler Films Production and MGM. Stallone will serve as writer in addition to starring in the film. Conceptualized as an [[epilogue]] story, the film is said to be about Rocky befriending a young fighter who is a foreigner, stuck illegally in the United States. Stallone states: &quot;Rocky meets a young, angry person who got stuck in this country when he comes to see his sister. He takes him into his life, and unbelievable adventures begin, and they wind up south of the border. It's very, very timely&quot;.&lt;ref&gt;{{cite web|url=https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/ |title=Sylvester Stallone Feels Robbed of an Ownership Stake in 'Rocky': 'I Was Furious' |work= [[Variety (magazine)|Variety]] |date=July 23, 2019 |access-date=July 23, 2019}}&lt;/ref&gt; By May 2020, Stallone said that he is still working on the film, though it has not yet been officially [[Green-light|green-lit]] by the studio.&lt;ref name=&quot;Rocky epilogue_Comicbook.com&quot;&gt;{{cite web |url=https://comicbook.com/movies/news/sylvester-stallone-new-rocky-movie-ideas-wont-appear-creed-III-michael-b-jordan/ |title=Sylvester Stallone Updates New Rocky Movie, Doesn't Expect to Appear in 'Creed III' With Michael B. Jordan |website=Comicbook.com |last=Bonomolo |first=Cameron |date=May 4, 2020 |access-date=January 16, 2021 }}&lt;/ref&gt; In November 2021, Stallone expressed doubt about the film being greenlit, due to his increasingly sour relationship with Irwin Winkler.&lt;ref&gt;{{Cite web |date=2021-11-02 |title=Stallone on the status of Rocky 7 |url=https://moviehole.net/stallone-on-the-status-of-rocky-7/ |access-date=2022-10-27 |website=Moviehole |language=en-US }}&lt;/ref&gt; In November 2022, Stallone confirmed that the studio wants another ''Rocky'' film, but that negotiations to attain part of the rights to the character from the producers stalled development. He further stated that he is writing the script and that if the studio likes his work, the film will be made.&lt;ref name=&quot;Future_THR&quot;&gt;{{cite web |date=November 7, 2022 |last=Hibberd |first=James |url=https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/ |work=[[The Hollywood Reporter]] |title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot; |access-date=November 8, 2022 }}&lt;/ref&gt;



1099 On December 13, 2006, it was officially announced by [[Ubisoft]] and MGM that a new ''Rocky'' video game, titled ''[[Rocky Balboa (video game)|Rocky Balboa]]'', was to be made exclusively for the [[PlayStation Portable]] handheld console. It was released on March 20, 2007, to coincide with the Blu-ray and DVD release.&lt;ref name=&quot;PSPRocky&quot;&gt;{{cite news|last=Luce|first=Patrick|url=https://www.monstersandcritics.com/dvd/news/article_1264409.php/Sylvester_Stallone%92s_Rocky_Balboa_comes_to_DVD_PSP_and_Blu-ray_in_March|archive-url=https://archive.today/20130129005909/http://www.monstersandcritics.com/dvd/news/article_1264409.php/Sylvester_Stallone%92s_Rocky_Balboa_comes_to_DVD_PSP_and_Blu-ray_in_March|url-status=dead|archive-date=January 29, 2013|title=Sylvester Stallone's Rocky Balboa comes to DVD, PSP and Blu-ray in March|publisher=Monsters and Critics|date=February 15, 2007|access-date=June 15, 2009}}&lt;/ref&gt;

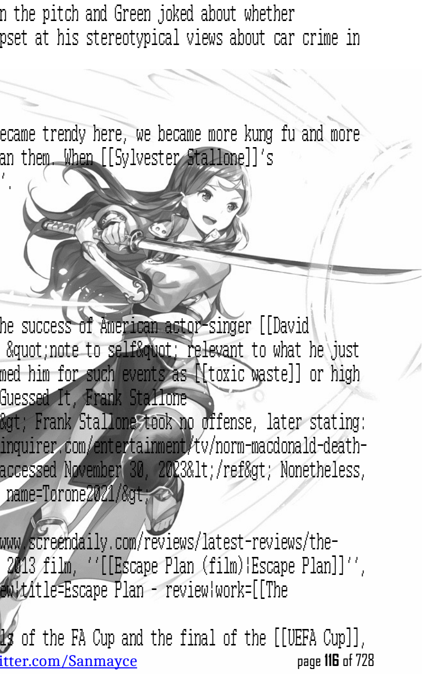
```
1100 {{Sylvester Stallone}}
1101 [[Category:Films directed by Sylvester Stallone]]
1102 [[Category:Films with screenplays by Sylvester Stallone]]
```

1103 However, advancements in [[digital compositing]] and the increasing use of digital cameras have made digital the most common method of choice. The last major [[blockbuster (entertainment)|blockbuster]] to extensively use front projection was the [[Sylvester Stallone]] [[Action film|action thriller]] ''[[Cliffhanger (film)|Cliffhanger]]'' from [[1993 in film|1993]].{{citation needed|date=December 2012}} More recently, the film ''[[Oblivion (2013 film)|Oblivion]]'' made extensive use of front projection (though not retro-reflective) to display various sky backgrounds in the home set. ''[[Spectre (2015 film)|Spectre]]'' also used this technique for its snow mountain hospital and glass building interiors. The advantages for the in-camera effect were a reduced need for digital effects and green screen, interactive lighting in a reflective set, and to provide a real background for the actors.

1104 * [[Animal Crackers (2017 film)|''Animal Crackers'' (2017 film)]], an animated movie starring Ian McKellen, Danny DeVito, and Sylvester Stallone

1105 In 1973, soon after the first film came out, producer [[Irwin Allen]] proposed a sequel that would have had the survivors testifying in a hearing on the disaster in [[Austria]], the country of the ''Poseidon''{{'}}s parent company. While on a train to the hearing, a miles-long mountain tunnel would collapse, leaving the survivors of the train trapped inside, struggling to make their way out. The film was planned to be released at Christmas 1974 from [[20th Century Fox]]. Most of the main cast was initially intended to reprise their roles from the first film.&lt;ref name=&quot;AFI&quot; /&gt; This premise was eventually used in the [[Rob Cohen]] film ''[[Daylight (1996 film)|Daylight]]'' (1996) with [[Sylvester Stallone]].&lt;ref&gt;From Special Edition DVD

```
     ''Beyond the Poseidon Adventure'' actor commentary.&lt;/ref&gt;
1106 The Robert DuBois incarnation of Bloodsport made his live-action debut in the television series ''[[Supergirl (TV series)|Supergirl]]'', played by David St. Louis.&lt;ref&gt;{{cite web|url=https://www.inverse.com/entertainment/the-suicide-squad-bloodsport-idris-elba|title=IDRIS ELBA'S THE SUICIDE SQUAD CHARACTER HINTS AT AN ALTERNATE DCEU TIMELINE|last=Abdulbaki|first=Mae|website=[[Inverse (website)|Inverse]]|date=August 25, 2020|access-date=January 1, 2021}}&lt;/ref&gt; [[Idris Elba]] portrayed the character in the [[DC Extended Universe]] film ''[[The Suicide Squad (film)|The Suicide Squad]]'' (2021).&lt;ref&gt;{{cite web|url=https://deadline.com/2020/11/the-suicide-squad-adds-sylvester-stallone-james-gunn-confims-on-instagram-1234615905/|title='The Suicide Squad' Adds Sylvester Stallone To Its Lineup, Confirmed By James Gunn On Instagram|last=Haring|first=Bruce|website=[[Deadline Hollywood]]|date=November 14, 2020|access-date=January 1, 2021}}&lt;/ref&gt; The Alexander Trent incarnation appeared in ''[[Justice League vs. the Fatal Five]]'', voiced by [[Tom Kenny]].
1107 For most of 2009, Barrell researched and wrote the ''Did You Know?'' page for the ''Sunday Times Magazine'', which included investigations into [[urban myth]]s, unsung heroes, and fictional characters based on real people. From June 2005 to January 2006, Barrell wrote the ''Sunday Times Magazine'' column &quot;Born on the Same Day&quot;, which compared and contrasted the lives of famous people with exactly the same birth date - such as [[Margaret Thatcher]] and [[Lenny Bruce]], [[Sylvester Stallone]] and [[George W. Bush]], [[Marc Bolan]] and [[Rula Lenska]], and [[Michael Jackson]] and [[Lenny Henry]].
1108 * In fall 2018, ''[[Escape Plan: The Extractors]]'' with [[Sylvester Stallone]] was filmed at the facility.&lt;ref&gt;{{Cite web|url=https://www.news-herald.com/news/mansfield-reformatory-still-grips-fans-years-after-movie-s-release/article_2ebd48af-3347-5d93-ba26-91398fde8e0e.html|title=Mansfield Reformatory still grips fans 25 years after movie's release|first=Clint |last=O'Connor |agency=Associated Press|website=The News-Herald|language=en|access-date=2019-03-11}}&lt;/ref&gt;
1109 After the credits, the scene shifts to Laurie's apartment where the Clone is shaving in the bathroom, ranting about an idea he has for the ultimate [[Rambo (film series)|Rambo]]/[[Rocky (film series)|Rocky]] sequel that would involve [[Sylvester Stallone]] playing both roles in a fight to the finish that would end both series. Afterward, the Clone joins Laurie in the living room and starts work on a song, lighting cigarette after cigarette. Laurie, however, would rather read a newspaper and complain about the Clone smoking too much. At one point Laurie says, &quot;You know we have to get this done by the benefit tonight.&quot; &quot;What you mean, we?&quot; the Clone replies sullenly.
1110 In the training montage sequence of the blockbuster 1976 film ''[[Rocky]]'', [[Rocky Balboa]], played by [[Sylvester Stallone]], can be seen running past ''Moshulu'' while training for his heavyweight championship bout against [[Apollo Creed]].{{needs reference|date=May 2024|reason=previous ref is a fan site}}
1111 [[Ellen Albertini Dow]] obtained her first screen credit when she was sixty-eight.&lt;ref&gt;{{cite web|url=http://www.fametracker.com/hey_its_that_guy/albertini_ellen_dow.shtml |archive-url=https://web.archive.org/web/20030225012149/http://www.fametracker.com/hey_its_that_guy/albertini_ellen_dow.shtml |url-status=dead |archive-date=2003-02-25 |title=Fametracker :: Hey! It's That Guy! :: Ellen Dow Albertini |access-date=2019-03-12}}&lt;/ref&gt; [[Rodney Dangerfield]] was an actor/comedian who did not really start until he was forty-two. He had done clubs when he was younger, but stopped in order to work as a salesman.&lt;ref&gt;{{cite news |url=http://www.msnbc.msn.com/id/6187136/|archive-url=https://web.archive.org/web/20041010040243/http://www.msnbc.msn.com/id/6187136/|url-status=dead|archive-date=10 October 2004|title=Rodney Dangerfield dead at 82|agency=Associated Press|publisher=MSNBC|date=7 October 2004|access-date=3 January 2009}}&lt;/ref&gt; [[Zelda Rubinstein]] was forty-eight before she had her first role, a minor part in ''[[Under the Rainbow]]'', but is more known for her &quot;debut&quot; in the [[Poltergeist (film series)|''Poltergeist'' film series]] starting the following year. Chicago native [[Chi McBride]], best known for the role as the principal in the series ''Boston Public'', only got into acting when he was thirty-one.&lt;ref&gt;{{IMDb name|id=0564277|name=Chi McBride}}&lt;/ref&gt; [[Danny Glover]] had a brief stint in the career of politics before he had involved himself in acting at twenty-eight.&lt;ref&gt;{{IMDb name|id=0000418|name=Danny Glover}}&lt;/ref&gt; BAFTA winning British actress [[Liz Smith (actress)|Liz Smith]] did not become a professional actress until the age of fifty. [[Kathryn Joosten]] also got a late start, beginning acting at age forty-two in community theater.&lt;ref&gt;{{IMDb name|0429760|name=Kathryn Joosten}}&lt;/ref&gt; Television star [[Judd Hirsch]] from ''Taxi'' and character actor [[Bill Cobbs]] became active at the age of thirty-six. [[George Wendt]] who played Norm on ''[[Cheers]]'' became active at the age of thirty-two. [[Brian Dennehy]] had dreams of stage and screen at an early age, but chose to first pursue other interests such as service in the U.S. Marine Corps prior to becoming active at the age of thirty-eight. Irish actor [[Brendan Gleeson]], who appeared as [[Mad Eye Moody]] in the ''[[Harry Potter (film series)|Harry Potter]]'' films and alongside [[Colin Farrell]] in ''[[In Bruges]]'', started acting professionally at thirty-four, having previous work as a school teacher. The Indian (Bengali) actor [[Paran Bandopadhyay]] is another late bloomer, who started his acting career in television and films at the age of sixty, after retiring from his government job. [[Jerry Doyle]], of ''[[Babylon 5]]'' fame, did not start acting until he was thirty-six after working as a stockbroker and pilot. [[Sylvester Stallone]] was thirty when he wrote and starred in the first ''[[Rocky]]''. All throughout his life, he has pushed his body through rigorous training routines for his film roles. Most notably at age forty-three, he developed his now-famous ''[[Rambo 3]]'' physique which got him named as &quot;body of the '80s&quot;. The veteran Indian (Malayalam) actor [[Sathyan (Malayalam actor)|Sathyan]] started his career at the age of forty, after resigning from the Police Service. He later came to be known as one of the greatest Malayalam film actors, and the actor [[Thilakan]] did not have much success until he was well into his forties, despite having started his film career at the age of thirty-seven after having worked in a drama troupe.
1112 In April 2016, Thompson was cast as [[Valkyrie (Marvel Comics)|Valkyrie]]&lt;ref name=&quot;Cast&quot;&gt;{{cite web|url=http://marvel.com/news/movies/26203/marvel_studios_confirms_stellar_new_cast_members_of_the_highly_anticipated_thor_ragnarok|title=Marvel Studios Confirms Stellar New Cast Members of the Highly Anticipated 'Thor: Ragnarok'|publisher=[[Marvel.com]]|date=May 20, 2016| first= Marc |last=Strom|access-date=May 20, 2016| archive-date= April 10, 2017| archive-url = https://web.archive.org/web/20170410160821/https://news.marvel.com/movies/26203/marvel_studios_confirms_stellar_new_cast_members_of_the_highly_anticipated_thor_ragnarok/ |url-status = live}}&lt;/ref&gt; in the [[Marvel Cinematic Universe]] superhero film ''[[Thor: Ragnarok]]'', which was released on November 3, 2017.&lt;ref&gt;{{cite web|url=https://deadline.com/2016/04/tessa-thompson-joins-the-marvel-universe-thor-ragnarok-1201735551/|title=Tessa Thompson Joins The Marvel Universe In 'Thor: Ragnarok'|website=[[Deadline Hollywood]]|last=Busch|first=Anita|date=April 11, 2016|access-date= April 10, 2017 |archive-date=February 23, 2017 | archive-url=https://web.archive.org/web/20170223113836/http://deadline.com/2016/04/tessa-thompson-joins-the-marvel-universe-thor-ragnarok-1201735551/ |url-status = live}}&lt;/ref&gt; She appeared in the science fiction horror film ''[[Annihilation (film)|Annihilation]]'', which was released on February 23, 2018. In June 2017, Thompson was cast in the science fiction comedy ''[[Sorry to Bother You]]'',&lt;ref&gt;{{cite web|url=https://deadline.com/2017/06/tessa-thompson-lakeith-stanfield-steven-yeun-movie-sorry-to-bother-you-1202113786/|title=Tessa Thompson, Lakeith Stanfield, Steven Yeun To Star In 'Sorry To Bother You'|last=Busch|first=Anita|website=[[Deadline Hollywood]]|date=June 15, 2017|access-date=May 19, 2018|archive-url=https://web.archive.org/web/20180612164652/https://deadline.com/2017/06/tessa-thompson-lakeith-stanfield-steven-yeun-movie-sorry-to-bother-you-1202113786/|archive-date=June 12, 2018|url-status=live}}&lt;/ref&gt; which was released on July 6, 2018.&lt;ref&gt;{{cite magazine|url=https://www.rollingstone.com/movies/news/watch-lakeith-stanfield-in-wild-sorry-to-bother-you-trailer-w520411|title=Watch Lakeith Stanfield Make Millions With 'White Voice' in Wild 'Sorry to Bother You' Trailer|last=Blistein|first=Jon|magazine=[[Rolling Stone]]|date=May 17, 2018|access-date=May 19, 2018|archive-url=https://web.archive.org/web/20180519204721/https://www.rollingstone.com/movies/news/watch-lakeith-stanfield-in-wild-sorry-to-bother-you-trailer-w520411|archive-date=May 19, 2018|url-status=live}}&lt;/ref&gt; She reprised her role as [[List of Rocky characters#Bianca Taylor|Bianca Taylor]] in the sports drama sequel film ''[[Creed II]]'', which was released on November 21, 2018, to positive reviews and strong box office returns.&lt;ref&gt;{{Cite news|url=https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/|title=Steven Caple Jr. to Direct 'Creed 2' Starring Michael B. Jordan and Sylvester Stallone (EXCLUSIVE)|last=Kroll|first=Justin|date=December 11, 2017|work=Variety|access-date=April 18, 2018|language=en-US|archive-url=https://web.archive.org/web/20190412030538/https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/|archive-date=April 12, 2019|url-status=live}}&lt;/ref&gt;
1113 | style=&quot;text-align: center;&quot;  | &lt;ref&gt;{{Cite news|url=https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/|title=Steven Caple Jr. to Direct 'Creed 2' Starring Michael B. Jordan and Sylvester Stallone (EXCLUSIVE)|last=Kroll|first=Justin|date=December 11, 2017|work=Variety|access-date=June 28, 2020|language=en-US|archive-url=https://web.archive.org/web/20190412030538/https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/|archive-date=April 12, 2019|url-status=live}}&lt;/ref&gt;
1114 In 2007, he commentated on a match between Everton and [[Reading F.C.|Reading]] at [[Goodison Park]]. Film star [[Sylvester Stallone]] was paraded on the pitch and Green joked about whether Stallone's limousine would still have wheels when he returned to it. This prompted an official complaint to the BBC by [[Liverpool City Council]], upset at his stereotypical views about car crime in the city.&lt;ref&gt;[http://www.radiotoday.co.uk/news.php?extend.1518 Radio Today with United Radio..: Five Live upsets Liverpool] {{webarchive |url=https://web.archive.org/web/20070927022002/http://www.radiotoday.co.uk/news.php?extend.1518 |date=27 September 2007 }}&lt;/ref&gt;
1115 * The Powerbook 540c is used by both  [[Antonio Banderas]] and [[Sylvester Stallone]] in the movie ''[[Assassins (1995 film)]]''.
1116 | quote = Copying is Philippine cinema's biggest crime even back then. In truth, it already became an art form here. When the Chinese' [[kung fu]] became trendy here, we became more kung fu and more Chinese than them. When the cowboy of Americans and Italians became trendy here, we became more [[Jesse James]] and [[Django (1966 film)|Django]] than them. When [[Sylvester Stallone]]'s ''[[Rocky]]'' was released, our ''Rocky Tulog'' and ''Totoy Bato'' sprung up. For ''[[Rambo (franchise)|Rambo]]'', our carbon copy is ''Rambo Tango''.
1117 * [[Sylvester Stallone]]
1118 [[Category:Images of Sylvester Stallone]]</text>
1119 [[Category:Images of Sylvester Stallone]]
1120 [[Category:Images of Sylvester Stallone]]</text>
1121 [[Category:Images of Sylvester Stallone]]</text>
1122 His version of ''Weekend Update'' often included [[running joke]]s about [[prison rape]], &quot;[[Drugs and prostitution|crack whores]]&quot;, and the success of American actor-singer [[David Hasselhoff]] in Germany. Macdonald would occasionally deliver a piece of news before taking out his personal compact [[tape recorder]] and leaving a &quot;note to self&quot; relevant to what he just discussed. He commonly used actor-singer [[Frank Stallone]] as a [[Non sequitur (absurdism)|non sequitur]] [[Punch line|punchline]] and absurdly blamed him for such events as [[toxic waste]] or high [[unemployment]] rates.&lt;ref name=&quot;stallone&quot;&gt;{{cite web |last1=Prigge |first1=Matt |title=Norm Macdonald Got Sweet Tribute From, You Guessed It, Frank Stallone |url=https://uproxx.com/tv/norm-macdonald-frank-stallone-tribute/ |website=UPROXX |access-date=September 15, 2021 |date=September 14, 2021}}&lt;/ref&gt; Frank Stallone took no offense, later stating: &quot;He wasn't really attacking me, it was just randomly thrown in there&quot;.&lt;ref name=Torone2021&gt;Rob Torone (Sep. 15, 2021). [https://www.inquirer.com/entertainment/tv/norm-macdonald-death-frank-stallone-saturday-night-live-snl-20210915.html Frank Stallone quietly loved being the butt of Norm Macdonald's jokes]. Philadelphia Inquirer, accessed November 30, 2023&lt;/ref&gt; Nonetheless, Macdonald stopped the Frank Stallone jokes after a 1997 request from [[Sylvester Stallone]], Frank's brother, who was guest host for ''SNL''.&lt;ref name=Torone2021/&gt;
1123 ** ''[[Get Carter (2000 film)|Get Carter]]'' (2000) with [[Sylvester Stallone]]
1124 He directed ''[[The Rite (2011 film)|The Rite]]'', an exorcism thriller film starring [[Anthony Hopkins]], in 2011.&lt;ref&gt;{{cite web|url=http://www.screendaily.com/reviews/latest-reviews/the-rite/5022755.article|title=The Rite - Reviews - Screen|work=[[Screen International]]|first=John|last=Hazelton|date=25 January 2011}}&lt;/ref&gt; His 2013 film, ''[[Escape Plan (film)|Escape Plan]]'', starred [[Arnold Schwarzenegger]] and [[Sylvester Stallone]].&lt;ref&gt;{{cite web|url=https://www.theguardian.com/film/2013/oct/17/escape-plan-review|title=Escape Plan - review|work=[[The Guardian]]|first=Mike|last=McCahill|date=17 October 2013}}&lt;/ref&gt;
1125 In [[1980-81 Ipswich Town F.C. season|1980-81]] Cooper made 61 appearances as Ipswich reached the [[EFL Cup|League Cup]] fourth round, the semi-finals of the FA Cup and the final of the [[UEFA Cup]],
```



in which they defeated [[AZ Alkmaar]] 5-4 on aggregate to win the club's first European trophy. However, they missed out on the league title after losing seven of their last ten matches, something that Cooper has said is the only disappointment of his career.&lt;ref name=GU&gt;[http://www.greenun24.co.uk/ipswich-town/ipswich_town_hall_of_fame_paul_cooper_reflects_on_the_glory_years_1_3461984 Ipswich Town Hall of Fame: Paul Cooper reflects on the glory years]{{dead link|date=February 2024|bot=medic}}Green 'Un, 21 March 2014&lt;/ref&gt; At the end of the 1980-81 season Cooper was named [[Ipswich Town F.C. Player of the Year|Player of the Year]] by Ipswich supporters. During this time, Cooper was the only regular first team member at Ipswich not to play for his country,&lt;ref name=DH/&gt; although he was considered unlucky not to do so, facing competition from [[Peter Shilton]] and [[Ray Clemence]].&lt;ref name=BBC1/&gt; In 1981, he was also one of several Ipswich players to appear in the film [[Escape to Victory]], with Cooper used as a stand-in for [[Sylvester Stallone]], whose character played in goal.&lt;ref&gt;[http://www.insidefutbol.com/2010/09/22/escape-to-victory-the-films-footballing-cast-revisited/30184/ Escape to Victory: The Film's Footballing Cast Revisited] Inside Futbol, 22 September 2010&lt;/ref&gt; The [[1981-82 Football League|following season]] saw Ipswich finish as league runners-up for a second consecutive season, although Cooper missed 10 league games, Sivell playing in nine and [[John Jackson (footballer, born 1942)|John Jackson]] making his only Ipswich appearance in the other.&lt;ref&gt;[http://www.prideofanglia.com/page.php?page=seasons&amp;text=1981-82 Season 1981-82 Division 1] Pride of Anglia&lt;/ref&gt;

1126 Based on a short student film, ''The Telephone'', which he made while attending NYU Film School,&lt;ref&gt;https://tisch.nyu.edu/content/dam/tisch/film-tvs/FirsRun/Archives/1981.pdf {{Bare URL PDF|date=September 2022}}&lt;/ref&gt; Salwen was accepted into the American Film Institute's Director's Internship Program under the co-sponsorship of The American Academy of Motion Picture Arts and Sciences where he was placed to observe Sylvester Stallone direct&lt;ref&gt;https://www.imdb.com/title/tt0086361/fullcredits/?ref_=tt_cl_sm {{User-generated source|certain=yes|date=August 2022}}&lt;/ref&gt; ''[[Staying Alive (1983 film)|Staying Alive]]''.&lt;ref&gt;[https://www.variety.com/article/VR1117792717/ Hal Salwen Variety article]&lt;/ref&gt;&lt;ref&gt;https://www.imdb.com/name/nm0759410/?ref_=ttfc_fc_cr136 {{User-generated source|certain=yes|date=August 2022}}&lt;/ref&gt; Following that experience (which he said did not heavily influence his directing style)&lt;ref&gt;{{cite web | url=https://variety.com/1997/film/news/hal-selwen-1117792717/ | title=Hal Selwen | date=15 January 1997 }}&lt;/ref&gt; Salwen began a career in television commercials in New York City, working in various capacities, from production assistant to producer, while simultaneously writing original screenplays on spec.

1127 Sketches on the show frequently featured celebrity impersonations, including Malone's portrayals of [[Barbara Frum]], [[Ronald Reagan]], [[John Forsythe]], [[Joan Collins]], [[Maureen Forrester]] and [[Elizabeth II|Queen Elizabeth II]], and Sexton's impressions of [[Sylvester Stallone]] and [[Ginette Reno]].&lt;ref name=gonational/&gt;

1128 Rovin has written encyclopedias about [[popular culture]], including ''[[The Encyclopedia of Superheroes]]'' (Facts On File, 1985), ''[[The Encyclopedia of Super Villains]]'' (Facts On File, 1987) ''The Illustrated Encyclopedia of Cartoon Animals'' (Prentice Hall, 1991), and ''The Encyclopedia of Monsters'' (Checkmark Books, 1990). He has worked on biographical and film books on such performers as [[Kelsey Grammer]], [[Lana Turner]], [[Adam West]], [[Ellen DeGeneres]], [[Jackie Chan]], [[Charlton Heston]], [[Elvis Presley]], [[Sylvester Stallone]], [[Richard Pryor]], [[Luke Perry]], [[Jason Priestley]], and [[Julio Iglesias]], and on the animated series ''[[The Simpsons]]''. Additionally, he has written quiz and joke books.

1129 *''[[Sylvester Stallone|Stallone]]!: A Hero's Story An Unauthorized Biography'' (1987)

1130 | {{won}}{{efn|Tied with [[Sylvester Stallone]] and [[Sharon Stone]] for ''[[The Specialist]]''.}}

1131 He appeared in the feature films, ''[[Life Happens]]'' (2011), ''[[Open Road (2012 film)|Open Road]]'' (2013), ''[[A Stranger in Paradise]]'' (2013), ''[[Vice (2015 film)|Vice]]'' with [[Bruce Willis]] (2015), ''[[Bad Moms]]'' (2016), ''[[The Space Between Us (film)|The Space Between Us]]'' with [[Gary Oldman]] (2017), ''[[Reprisal (2018 film)|Reprisal]]'' (2018), ''[[Backtrace (film)|Backtrace]]'' opposite [[Sylvester Stallone]] (2018), ''[[100 Days to Live]]'' (2019), and in Lifetime's ''[[A Rose for Her Grave and Other True Cases|A Rose for Her Grave]]''. He has appeared in numerous television shows including ''[[Hawaii Five-0 (2010 TV series)|Hawaii Five-O]]'', ''[[Chicago Fire (TV series)|Chicago Fire]]'', ''[[Lucifer]]'' and ''[[Unreal (TV series)|UnReal]]'', and ''[[Gilmore Girls]]''.

1132 | father   = [[Sylvester Stallone]]

1133 '''Sage Moonblood Stallone''' (May 5, 1976 - July 13, 2012)&lt;!--Do not change date without discussion.--&gt; was an American actor. He was the eldest child of actor [[Sylvester Stallone]].&lt;ref&gt;{{cite web|url=https://movies.yahoo.com/person/sage-stallone/|title=Sage Stallone|access-date=June 9, 2012|publisher=Yahoo!}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://www.rottentomatoes.com/celebrity/sage_stallone/biography.php |title=Sage Stallone Biography|access-date=June 9, 2012|work=[[Rotten Tomatoes]] | publisher= [[Flixster]] ([[Time Warner]])}}&lt;/ref&gt;&lt;ref&gt;{{cite news|title= Stallone Jr. Hopes Playing Rocky Jr. Won't Cramp His Lifestyle|work=[[Los Angeles Times]]| url= https://www.latimes.com/archives/la-xpm-1990-11-24-ca-4493-story.html|access-date=October 28, 2010}}&lt;/ref&gt;

1134 Sage Stallone was born in [[Los Angeles|Los Angeles, California]], the elder son and first child of Sasha Czack and actor [[Sylvester Stallone]]. He was the brother of Seargeoh &quot;Seth&quot; Stallone, and half-brother of Sophia, [[Sistine Stallone|Sistine]], and Scarlet Stallone. He was the nephew of actor and singer [[Frank Stallone]], and grandson of [[Jackie Stallone]]. His stepmother was the model and entrepreneur [[Jennifer Flavin]].

1135 Stallone made his acting debut alongside his father in ''[[Rocky V]]'' (1990), the fifth installment of the [[Rocky (film series)|''Rocky'' franchise]], playing Robert Balboa Jr., the onscreen son of his father's title character. He also appeared with his dad in ''[[Daylight (1996 film)|Daylight]]'' (1996).&lt;ref&gt;{{cite web|url=https://people.com/celebrity/sage-stallone-sylvester-stallone-heartache-and-reconciliation/|title=Sage and Sylvester Stallone: A Story of Heartache and Reconciliation|last=Leonard|first=Elizabeth|last2=Nudd|first2=Tim |date=July 17, 2012|work=People|access-date=February 28, 2019}}&lt;/ref&gt; He subsequently appeared in low budget exploitation films.

1136 In 1996, Stallone and [[film editor]] [[Bob Murawski]] co-founded [[Grindhouse Releasing]], a Los Angeles-based company dedicated to the restoration and preservation of [[exploitation film]]s such as ''[[Cannibal Holocaust]]'' and ''[[Gone with the Pope]]''.&lt;ref name=&quot;la times obit&quot;&gt;{{citation | url= http://www.latimes.com/news/obituaries/la-me-0714-sage-stallone-20120714,0,3532003.story |title= Sage Stallone dies at 36; son of Sylvester Stallone | first= Bob | last= Pool | work= Los Angeles Times | date= July 14, 2012}}&lt;/ref&gt;

1137 Stallone was found dead on July 13, 2012, at his home in [[Studio City, Los Angeles]].&lt;ref name=&quot;la times obit&quot; /&gt; He was 36 years old. According to reports, he had not been heard from for four days prior to his death.&lt;ref&gt;{{cite web|url=http://www.foxnews.com/entertainment/2012/07/16/sylvester-stallone-begs-for-privacy-following-sudden-death-his-son-sage/?intcmp=features|title=Sylvester Stallone begs for privacy following the sudden death of his son Sage|publisher=FoxNews.com|access-date=16 July 2012}}&lt;/ref&gt; An autopsy by the Los Angeles coroner and toxicology tests determined that Stallone died of [[coronary artery disease]] caused by [[atherosclerosis]], with no drugs detected other than an over-the-counter pain remedy.&lt;ref&gt;{{cite web |last1=Duke |first1=Alan |title=Sage Stallone died of heart attack, not drugs, coroner concludes |url=https://www.cnn.com/2012/08/30/showbiz/sage-stallone-death/ |website=CNN |access-date=August 14, 2022 |date=August 31, 2012}}&lt;/ref&gt; At the time of his death, Stallone was reportedly engaged.&lt;ref name=&quot;la times obit&quot; /&gt;&lt;ref&gt;{{cite web|url=http://www.people.co.uk/news/uk-world-news/2012/07/15/sylvester-stallone-in-state-of-shock-over-son-sage-s-drugs-overdose-death-102039-23908644/|title=Sylvester Stallone in state of shock over son Sage's drugs overdose death|last=Panther|first=Lewis|date=July 15, 2012|work=People UK|access-date=March 2, 2013}}&lt;/ref&gt; Stallone's funeral was held on July 21 at [[St. Martin of Tours Catholic Church]] in Los Angeles.&lt;ref&gt;{{cite web|url=https://people.com/celebrity/sylvester-stallones-son-sage-stallone-laid-to-rest/|title=Sage Stallone Laid to Rest|date=July 21, 2012|work=People|access-date=February 28, 2019}}&lt;/ref&gt; He is interred at [[Westwood Village Memorial Park Cemetery]].&lt;ref&gt;{{cite web |last1=Meares |first1=Hadley |title=Eternity with the In-Crowd: Funny Fellows, Doomed Damsels and Hollywood Hangers-on of Westwood Village Memorial Park |url=https://www.kcet.org/history-society/eternity-with-the-in-crowd-funny-fellows-doomed-damsels-and-hollywood-hangers-on-of-westwood-village-memorial-park |website=KCET |access-date=August 14, 2022 |date=October 25, 2013}}&lt;/ref&gt;

1138 [[Category:Family of Sylvester Stallone]]

1139 He is the official dub-over artist of [[Arnold Schwarzenegger]] and has met with Schwarzenegger several times.&lt;ref&gt;{{Cite web |author = |date = 8 July 2015|url = https://eiga.com/news/20150708/10/|title = シュワ声優・玄田哲章、初対面の本人から日本語版吹き替え声優に永久指名! |website = eiga.com|accessdate = 14 October 2020}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://news.mynavi.jp/article/20191107-920417/|title = 来日中の A・シュワルツェネッガーが歌舞伎町に登場「また戻ってきました!」|date = 7 November 2019}}&lt;/ref&gt; He is a standard choice for Japanese dubbing of the voices of such English-speaking actors as [[Samuel L. Jackson]], [[John Goodman]], [[Dan Aykroyd]], [[Laurence Fishburne]], and [[Gérard Depardieu]]. He was the first dubbing actor for [[Sylvester Stallone]] and [[Steven Seagal]] in their early days.{{citation needed|date=May 2023}}

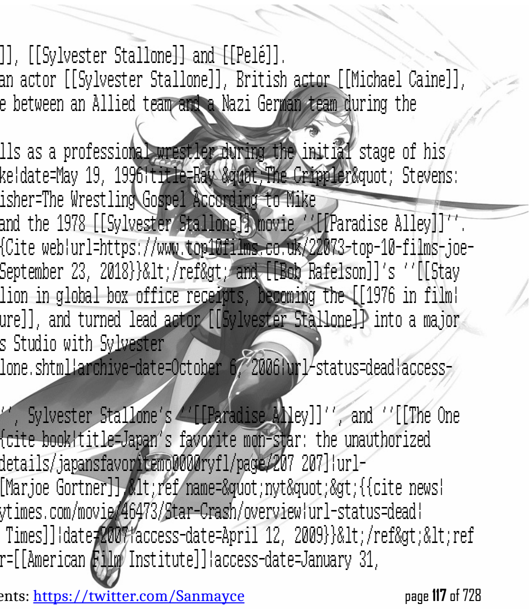
1140 *'''[[Sylvester Stallone]]'''

1141 **''[[An Alan Smithee Film: Burn Hollywood Burn]]'' (Sylvester Stallone)&lt;!--1997--&gt;

1142 **''[[Taxi 3]]'' (Sylvester Stallone)&lt;!--2003--&gt;

1143 O'Callaghan appeared in the 1981 movie ''[[Escape to Victory]]'', cast in the unfamiliar role of goalkeeper alongside [[Michael Caine]], [[Sylvester Stallone]] and [[Pelé]].

1144 He played Doug Clure, one of the prisoner-of-war footballers, in the 1981 film ''[[Escape to Victory]]'', which also included American actor [[Sylvester Stallone]], British actor [[Michael Caine]], Swedish actor [[Max von Sydow]], and football stars [[Pelé]], [[Ossie Ardiles]], and [[Kazimierz Deyna]]. The film was based on a game between an Allied team and a Nazi German team during the [[Second World War]].



1145 In 1952, Stevens was married to female wrestler Theresa Theis, who also did some work as Stevens's trainer and helped to hone his skills as a professional wrestler during the initial stage of his career.&lt;ref name=&quot;:0&quot; /&gt; They were divorced in 1972.&lt;ref name=&quot;:3&quot;&gt;{{Cite web|last=Mooneyham|first=Mike|date=May 19, 1996|title=Ray &quot;The Crippler&quot; Stevens: Bombs Away|url=http://www.mikemooneyham.com/1996/05/19/ray-the-crippler-stevens-bombs-away/|access-date=August 17, 2021|website=|publisher=The Wrestling Gospel According to Mike Mooneyham}}&lt;/ref&gt;&lt;ref name=&quot;:7&quot; /&gt; He appeared in the 1974 movie ''[[The Wrestler (1974 film)|The Wrestler]]'' and the 1978 [[Sylvester Stallone]] movie ''[[Paradise Alley]]''.

1146 In 1976, he acted in [[Paul Mazursky]]'s ''[[Next Stop, Greenwich Village]],'' [[Martin Scorsese]]'s ''[[Taxi Driver]]'',&lt;ref&gt;{{Cite web|url=https://www.top10films.co.uk/22073-top-10-films-joe-spinell/|title=Top 10 Films Of Joe Spinell - Top 10 Films|website=www.top10films.co.uk|date=July 4, 2014 |language=en-GB|access-date=September 23, 2018}}&lt;/ref&gt; and [[Bob Rafelson]]'s ''[[Stay Hungry]].'' That year, Spinell played the role of Gazzo, a [[loan shark]] in [[John G. Avildsen]]'s ''[[Rocky]].'' It earned $225 million in global box office receipts, becoming the [[1976 in film|highest-grossing film of 1976]], went on to win three [[Academy Awards|Oscars]], including [[Academy Award for Best Picture|Best Picture]], and turned lead actor [[Sylvester Stallone]] into a major star.&lt;ref&gt;{{cite web|url=http://www.bravotv.com/Inside_the_Actors_Studio/guests/Sylvester_Stallone.shtml|title=Inside the Actors Studio with Sylvester Stallone|archive-url=https://web.archive.org/web/20061006041106/http://www.bravotv.com/Inside_the_Actors_Studio/guests/Sylvester_Stallone.shtml|archive-date=October 6, 2006|url-status=dead|access-date=September 28, 2006}}&lt;/ref&gt;

1147 In 1978, he acted in [[Paul Williams (director)|Paul Williams]]'s ''[[Nunzio (film)|Nunzio]]'', [[John Milius]]'s ''[[Big Wednesday]]'', Sylvester Stallone's ''[[Paradise Alley]]'', and ''[[The One Man Jury]]''. He also played the main antagonist in [[Luigi Cozzi]]'s Italian-produced [[space opera]] ''[[Starcrash]]'',&lt;ref&gt;{{cite book|title=Japan's favorite mon-star: the unauthorized biography of &quot;The Big G&quot;|last=Ryfle|first=Steve|publisher=ECW Press|year=1999|isbn=1-55022-348-8|page=[https://archive.org/details/japansfavoritemo0000ryfl/page/207 207]|url-access=registration|url=https://archive.org/details/japansfavoritemo0000ryfl/page/207}}&lt;/ref&gt; starring [[Caroline Munro]] and [[Marjoe Gortner]].&lt;ref name=&quot;nyt&quot;&gt;{{cite news|url=https://movies.nytimes.com/movie/46473/Star-Crash/overview|archive-url=https://web.archive.org/web/20071211100444/http://movies.nytimes.com/movie/46473/Star-Crash/overview|url-status=dead|archive-date=December 11, 2007|title=Star Crash (1978)|last=Wheeler|first=Jeremy|department=Movies &amp; TV Dept.|work=[[The New York Times]]|date=2007|access-date=April 12, 2009}}&lt;/ref&gt;&lt;ref name=&quot;afi&quot;&gt;{{cite web|url=http://www.afi.com/members/catalog/DetailView.aspx?s=&amp;Movie=57096|title=Starcrash|publisher=[[American Film Institute]]|access-date=January 31, 2016}}&lt;/ref&gt;&lt;ref name=&quot;mfb-review&quot;&gt;{{cite magazine|last=Pulleine|first=Tim|year=1979|title=Starcrash|magazine=[[Monthly Film Bulletin]]|location=London|publisher=[[British Film Institute]]|volume=46|issue=540|page=155}}&lt;/ref&gt;

1148 In 1979, he acted in [[Jonathan Demme]]'s ''[[Last Embrace]]'',&lt;ref name=&quot;Last Embrace&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/22824/Last-Embrace/|title=Last Embrace|work=[[Turner Classic Movies]]|access-date=March 9, 2016}}&lt;/ref&gt; and [[William Richert]]'s ''[[Winter Kills (film)|Winter Kills]].'' Spinell reprised his role as Gazzo in ''[[Rocky II]]'' this time directed by [[Sylvester Stallone]]. ''Rocky II'' finished in the top three highest-grossing films of 1979, in both the [[1979 in film#Highest-grossing films|North American market]] and [[List of highest-grossing films by year|worldwide]]. The film grossed $6,390,537 during its opening weekend, $85,182,160 at the U.S. box office, and $200,182,160 overall.&lt;ref name=&quot;The Numbers&quot;&gt;{{cite web|url=http://www.the-numbers.com/movies/1979/0RKY2.php|title=Movie Rocky 2 - Box Office Data, News, Cast Information|work=The Numbers|publisher=Nash Information Services|access-date=September 1, 2010}}&lt;/ref&gt;

1149 In 1981, Spinell had a supporting role in the Sylvester Stallone action film ''[[Nighthawks (1981 film)|Nighthawks]],''&lt;ref&gt;{{Cite web|url=https://catalog.afi.com/Catalog/moviedetails/56666|title=AFI{{!}}Catalog|website=catalog.afi.com|language=en|access-date=September 23, 2018}}&lt;/ref&gt; and [[Richard Elfman]]'s ''[[Forbidden Zone]].''

1150 A close friend of Sylvester Stallone, Spinell was the godfather of his son [[Sage Stallone]].&lt;ref&gt;{{Cite news|url=http://whatculture.com/film/sage-stallone-son-of-sylvester-found-dead-cause-of-death-still-unknown|title=Sage Stallone, Son of Sylvester, Found Dead - Cause of Death Still Unknown|date=July 15, 2012|work=WhatCulture.com|access-date=September 23, 2018|language=en}}&lt;/ref&gt; Spinell had a falling out with Sylvester Stallone during the filming of their final collaboration ''[[Nighthawks (1981 film)|Nighthawks]]'' (1981).&lt;ref&gt;{{Cite news|url=http://www.aintitcool.com/node/30932|title=Stallone answers December 9th &amp; 10th Questions in a double round - plus Harry's Seen ROCKY BALBOA...|last=headgeek|work=Aint It Cool News|access-date=September 30, 2018|language=en}}&lt;/ref&gt;

1151 | [[Sylvester Stallone]]

1152 In 1994 (see [[1994 in music]]), the song was recorded by Cuban-American singer and songwriter [[Gloria Estefan]] for the soundtrack to the film ''[[The Specialist]]'', starring [[Sylvester Stallone]] and [[Sharon Stone]]. It is also featured on Estefan's fourth solo album, ''[[Hold Me, Thrill Me, Kiss Me (Gloria Estefan album)|Hold Me, Thrill Me, Kiss Me]]'' (1994). Released as a single in September 1994 by [[Epic Records|Epic]], the song became a hit reaching number 13 on the [[Billboard Hot 100|''Billboard'' Hot 100]]. Estefan also took the song to the top spot on [[Billboard (magazine)|''Billboard'']]'s [[Hot Dance Club Play]] chart, making it her first number-one song on that chart in [[English language|English]]. In Australia and New Zealand the song charted at number eight and 18. In Europe, the song peaked at number 21 in the United Kingdom, number 24 in Scotland, number 27 in the Netherlands and number 29 in Belgium. Estefan performed the song as her opening performance in VH1's first ever [[VH1 Divas Live|Divas Live]]. In 2018, Australian music channel [[Max (Australian TV channel)|Max]] included Estefan's &quot;Turn the Beat Around&quot; in their list of &quot;1000 Greatest Songs of All Time&quot;.&lt;ref&gt;{{cite web|title= MAX's 1000 Greatest Songs Of All Time |publisher= [[Max (Australian TV channel)|Max]] |date= 2018 |access-date= 2020-04-26 |url= https://www.maxtv.com.au/the-2018-results}}&lt;/ref&gt;

1153 * [[Sylvester Stallone]]

1154 '''''Assassins''''' is a 1995 American [[action film|action]] [[thriller (genre)|thriller film]] directed by [[Richard Donner]]. It stars [[Sylvester Stallone]], [[Antonio Banderas]], and [[Julianne Moore]]. The screenplay was written by [[The Wachowskis]] and [[Brian Helgeland]]. The Wachowskis stated that their script was completely rewritten by Helgeland and that they tried to remove their names from the film but failed.&lt;ref&gt;[http://www.asitecalledfred.com/interviews/27.html] {{webarchive|url=https://web.archive.org/web/20120725034551/http://www.asitecalledfred.com/interviews/27.html|date=July 25, 2012}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine|url=http://www.newyorker.com/reporting/2012/09/10/120910fa_fact_hemon?currentPage=all |title=Beyond the Matrix |magazine=The New Yorker |date=2012-09-10 |access-date=2014-07-28}}&lt;/ref&gt;

1155 * [[Sylvester Stallone]] as Robert Rath/Joseph Rath

1156 * [[Sylvester Stallone]], born Sylvester Enzio Stallone (July 6, 1946, in New York City) American film actor, director, producer and screenwriter. He is often referred to by his nickname, &quot;Sly&quot;. He achieved his greatest successes in a number of [[action film]]s, notably the ''[[Rocky (film series)|Rocky]]'' and ''[[Rambo (film series)|Rambo]]'' series. He was born to Frank Stallone Sr. (a beautician who was an immigrant from [[Castellammare del Golfo]], Sicily) and Jacqueline &quot;Jackie&quot; Labofish, an American astrologer of 1/4 Russian Jewish descent.

1157 '''[[Sylvester Stallone]]'''

1158 Roderick Thorp decided for the book to be a sequel to ''The Detective'' so it could be made into a follow-up film starring [[Frank Sinatra]] as Joe Leland. Thus, the storyline hypothetically takes place some four decades after its predecessor (as Leland was age thirty-six in that novel). Sinatra declined the offer. It was then offered to [[Arnold Schwarzenegger]], [[Sylvester Stallone]] and a number of other actors until Bruce Willis signed on for the role.&lt;ref&gt;{{cite web|url=http://www.thedailybeast.com/articles/2016/06/29/die-hard-how-bruce-willis-changed-the-movies.html|title='Die Hard': How Bruce Willis Changed the Movies|publisher= The Daily Beast Archive|date=2012-06-29|accessdate=2016-12-08}}&lt;/ref&gt;

1159 Papale, who earned the nickname &quot;Rocky&quot; (after the [[Sylvester Stallone]] [[Rocky Balboa (character)|character]]) while playing with the Eagles, is the subject of the [[The Walt Disney Company|Disney]] movie ''[[Invincible (2006 film)|Invincible]]'', with [[Mark Wahlberg]] portraying him. In addition, Papale's legend was cited as a factor in the Eagles' signing of construction foreman/tight end [[Jeff Thomason]] for 2005's [[Super Bowl XXXIX]].&lt;ref&gt;{{cite web|url=http://www.philadelphiaeagles.com/homeNewsDetail.jsp?id=22443|title=Thomason Is Now A National Story - Philadelphia Eagles&lt;!-- Bot generated title --&gt;|access-date=August 16, 2006|archive-url=https://web.archive.org/web/20061108030638/http://www.philadelphiaeagles.com/homeNewsDetail.jsp?id=22443|archive-date=November 8, 2006|url-status=dead}}&lt;/ref&gt;

1160 | [[Sylvester Stallone]]

1161 | [[Sylvester Stallone]]

1162 When the awards reach the Best Actor award, Dr. Zoidberg tosses presenter [[Billy Crystal]] off the stage and takes his place. In place of the fifth nominee, he substitutes Calculon. But when he sees his uncle's depression at being a has-been, Zoidberg announces him as the winner. In his acceptance speech, Harold Zoid says his nephew's gesture has made him realize that the award itself is secondary to the knowledge that someone, even if only one person, still respects him as a filmmaker. Calculon, somewhat chastened by this speech, decides not to kill him or the others. Fry and Leela finally escape from the tar pits and are allowed to enter the after party when the skeleton they are dragging is recognized as that of [[Sylvester Stallone]].

1163 The character has been adapted from the comics into various forms of media, including television series, feature films, and video games. King Shark made his live-action debut in the television series ''[[The Flash (2014 TV series)|The Flash]]'', voiced by [[David Hayter]] while [[Dan Payne]] portrayed his human form. In the [[DC Extended Universe]], King Shark was voiced by actor [[Sylvester Stallone]] and [[motion capture]]d by [[Steve Agee]] in the film ''[[The Suicide Squad (film)|The Suicide Squad]]'' (2021).

1164 * [[King Shark (DC Extended Universe)|Nanaue / King Shark]] appears in ''[[The Suicide Squad (film)|The Suicide Squad]]'', voiced by [[Sylvester Stallone]]&lt;ref&gt;{{Cite web |last=Haring |first=Bruce |date=November 14, 2020 |title='The Suicide Squad' Adds Sylvester Stallone To Its Lineup, Confirmed By James Gunn On Instagram |url=https://deadline.com/2020/11/the-suicide-squad-adds-sylvester-stallone-james-gunn-confims-on-instagram-1234615905/ |url-status=live |archive-url=https://web.archive.org/web/20201114232924/https://deadline.com/2020/11/the-suicide-squad-adds-sylvester-stallone-james-gunn-confims-on-instagram-1234615905/ |archive-date=November 14, 2020 |access-date=November 14, 2020 |website=[[Deadline Hollywood]]}}&lt;/ref&gt; and [[motion capture]]d by [[Steve Agee]].&lt;ref&gt;{{cite news|url=http://collider.com/suicide-squad-2-new-characters/|title='Suicide Squad 2' Exclusive: Meet the New Characters James Gunn Will Introduce in Sequel|first=Jeff|last=Sneider|work=Collider|date=March 7, 2019|access-date=March 7, 2019}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Gonzalez |first=Umberto |date=May 9, 2019 |title=The Suicide Squad: Michael Rooker as King Shark in the New Film? |url=https://www.thewrap.com/michael-rooker-king-shark-suicide-squad/ |access-date=May 10, 2019 |website=Thewrap.com}}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=https://deadline.com/2019/08/suicide-squad-james-gunn-casts-steve-agee-king-shark-1202703447/|title=Comedian Steve Agee Joining James Gunn's 'Suicide Squad' Gang|first=Anthony|last=D'Alessandro|website=Deadline.com|date=August 24, 2019|access-date=August 24, 2019}}&lt;/ref&gt; This version displays a childlike demeanor and speaks in simple sentences. He is recruited into the titular team to infiltrate the [[Corto Maltese (DC Comics)|Corto Maltesean]] [[Nazi]]-era research facility Jötunheim and destroy &quot;[[Starro|Project Starfish]]&quot;, during which he befriends teammate [[Ratcatcher 2]].

1165 His most notable games include ''Cobra'',&lt;ref name='wireframe'&gt;{{cite magazine |title=Developer Profile: Special FX|magazine=Wireframe|issue=4|publisher=Raspberry Pi Ltd|date=10 October 2019|page=51|url=https://archive.org/details/Wireframe24/page/n49/mode/2up|access-date=9 June 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite journal |date=December 1986 | title = ''Cobra'' | journal = [[CRASH (magazine)|CRASH]] | issue = 35 | pages = 140-141 | url = http://www.crashonline.org.uk/35/cobra.htm | accessdate = 12 August 2006}}&lt;/ref&gt; (which was a license extremely loosely based on a [[Cobra (1986 film)|Sylvester Stallone film]]) one of the first Spectrum games to exhibit full-colour parallax scrolling and his conversion of the arcade game ''[[Rush N Attack|Green Beret]]''.&lt;ref name='wireframe' /&gt;&lt;ref&gt;{{cite journal | last = Smith | first = Rachael | date = June 1986 | title = ''Green Beret'' | journal = [[Your Sinclair]] | issue = 6 | pages = 30 | url = http://www.ysrnry.co.uk/articles/greenberet.htm | accessdate = 12 August 2006 | url-status = dead | archiveurl = https://web.archive.org/web/20060705103934/http://www.ysrnry.co.uk/articles/greenberet.htm | archivedate = 5 July 2006 }}&lt;/ref&gt; A notable &quot;trademark&quot; of Smith's was his habit of writing his name (with &quot;Jonathan&quot; and &quot;Smith&quot; spelled various ways) as a mirror-image in his games.

1166 ''[[Rocky]]'' was shot on a budget of $1 million and eventually grossed $225 million worldwide, making [[Sylvester Stallone]] a star.&lt;ref&gt;{{cite news|url=https://www.latimes.com/archives/la-xpm-2001-apr-26-ca-55716-story.html|title=Sly's Once-Rocky Life|last=King|first=Susan|work=[[Los Angeles Times]]|date=April 26, 2001|access-date=August 17, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.the-numbers.com/movie/Rocky#tab=summary|title=Rocky (1976)|work=[[The Numbers (website)|The Numbers]]|access-date=August 17, 2015}}&lt;/ref&gt; ''[[Halloween (1978 film)|Halloween]]'' was produced on a budget of $325,000 and grossed $70 million worldwide.&lt;ref&gt;{{cite web|url=https://deadline.com/2014/10/john-carpenter-qa-halloween-sequels-michael-myers-861942/|title=John Carpenter Q&amp;A: Why 'Halloween' Didn't Need Sequels &amp; What Scares The Master Of Horror|last=Yamato|first=Jen|website=[[Deadline Hollywood]]|date=October 31, 2014|access-date=August 17, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.the-numbers.com/movie/Halloween#tab=summary|title=Halloween (1978)|work=[[The Numbers (website)|The Numbers]]|access-date=August 17, 2015}}&lt;/ref&gt; ''[[Napoleon Dynamite]]'' cost less than $400,000 to make but its gross revenue was $46 million.&lt;ref&gt;{{cite web|url=https://wegotthiscovered.com/movies/10-profitable-low-budget-films/2/|title=Napoleon Dynamite|last=Lowe|first=Alexander|work=We Got This Covered|date=July 2, 2013|access-date=August 17, 2015}}&lt;/ref&gt; Divisions of [[major film studio]]s that specialize in such films, such as [[Fox Searchlight Pictures]], [[Miramax]], and [[New Line Cinema]], have made the distribution of low budget films competitive.&lt;ref&gt;{{cite book|title=American Independent Cinema: Indie, Indiewood and Beyond|last1=King|first1=Geoff|last2=Molloy|first2=Claire|last3=Tzioumakis|first3=Yannis|publisher=[[Routledge]]|year=2013|isbn=9780415684286|page=206|url=https://books.google.com/books?id=7ibUEeJMNCYC&amp;pg=PA206}}&lt;/ref&gt;

1167 In 1990, the three remaining members of the group premiered their [[sketch comedy]] television show, &quot;La Télé des Inconnus&quot;. The show, which ran for two years on Antenne 2 (now [[France 2]]), was highly successful, and propelled the group to stardom throughout France. Their sketches parodied a number of French cultural icons, highly popular at the time, including musicians [[Richard Clayderman]], [[Mano Negra (band)|Mano Negra]], [[Indochine (band)|Indochine]], [[Didier Barbelivien]], [[Florent Pagny]], and [[Patrick Bruel]], as well as international stars like [[Sylvester Stallone]] (in the unlikely role of [[Jesus|Jesus Christ]] in a mock film trailer), [[Jean-Claude Van Damme]] (a recurring character in unlikely roles like [[Jean Valjean]] from ''[[Les Misérables]]'' or Vicomte de Valmont from ''[[Les Liaisons dangereuses|Les liaisons dangereuses]]''), [[Al Pacino]], [[Robert De Niro|Robert de Niro]], and television shows ''[[Dorothée|Club Dorothée]]'', ''[[Choudenshi Bioman|Bioman]]'', ''[[Star Trek: The Original Series|Star Trek]]'' or ''[[Santa Barbara (TV series)|Santa Barbara]]''. They also satirized French politicians, police,

[[trade union]]s, [[contemporary art]], [[fashion design]], [[advertising]], various television programs, and society at large. The program won two ''[[7 d'Or]]'' (the top French television award), and a video clip from the show, &quot;[[Auteuil, Neuilly, Passy]]&quot;, won two ''[[Victoires de la Musique]]''. Their most famous sketch is probably &quot;''[[Télémagouilles]]''&quot;{{Citation needed¦date=January 2020}} with its gimmick &quot;[[Stéphanie de Monaco]]&quot;; this sketch is a parody of most of TV games.

1168 Budgets for their films grew. However, with rather poor marketing, Savoy faced a major financial slump, only three years after being formed. For three years, Savoy then released box office failures including ''[[Exit to Eden (film)¦Exit to Eden]]'' and ''[[Getting Away with Murder (1996 film)¦Getting Away with Murder]]''. It also didn't help that two of its competitors in the independent film field, [[Miramax]] and [[New Line Cinema]], were bought out by majors ([[The Walt Disney Company]] and [[Turner Broadcasting]], respectively), giving them stability. As a result, Savoy focused on low-budget films and the occasional blockbuster, costing up to $80 million.&lt;ref name=&quot;ser&quot;&gt;Fabrikant, Geraldine. [https://www.nytimes.com/1995/01/30/business/savoy-pictures-high-and-low-roads.html?src=pm ''Savoy Pictures' High and Low Roads''.] [[The New York Times]] (January 18, 1995)&lt;/ref&gt; Executives hoped to lure [[Sylvester Stallone]] with a then-hefty $20 million paycheck to star in a studio project that was ultimately never made.&lt;ref name=&quot;ser&quot; /&gt;

1169 While they were working to secure the film rights, Pressman and Summer were also contemplating the lead role. Summer said they considered [[Charles Bronson]], [[Sylvester Stallone]] and [[William Smith (actor)¦William Smith]]-all of whom had played tough figures{{Sfn¦Sammon¦2007¦p=100}} but, in 1976, the two producers watched a rough cut of the bodybuilding film, ''[[Pumping Iron]]'', and agreed that Arnold Schwarzenegger was perfect for the role of Conan due to his huge, muscular frame.{{Sfn¦Sammon¦2007¦p=101}} According to Schwarzenegger, Pressman's &quot;low-key&quot; approach and &quot;great inner strength&quot; convinced him to join the project.{{Sfn¦Turan¦1980¦p=63}} Paul Sammon, writer for ''[[Cinefantastique]]'', said that the former champion bodybuilder was practically the &quot;living incarnation of one of Frazetta's paperback illustrations&quot;.{{Sfn¦Sammon¦1982a¦p=30}} Schwarzenegger was paid $250,000 and placed on [[Retainer agreement¦retainer]];{{Sfn¦Andrews¦1995¦p=101}} the terms of the contract restricted him from starring in other sword-and-sorcery films.{{Sfn¦Gallagher¦Milius¦1989¦p=26}} Schwarzenegger said ''Conan'' was his biggest opportunity to establish himself in the entertainment industry.{{Sfn¦Turan¦1980¦pp=63, 66}}

1170 Celebrities whose autobiographies were read included Burt Reynolds, Loni Anderson, members of [['N Sync]], [[Madonna (entertainer)¦Madonna]], [[Sylvester Stallone]], [[Mr. T]], [[Elizabeth Taylor]], [[Kathie Lee Gifford]], [[Ivana Trump]], [[Zsa Zsa Gabor]], [[Vanna White]], [[Kenny Loggins]], [[Tommy Lee]] and [[David Cassidy]]. Comedians included [[Bruce Vilanch]], [[Jay Mohr]], [[Cheryl Hines]], [[Kevin Nealon]], [[Doris Roberts]], [[Fred Willard]], [[Niecy Nash]], [[Kel Mitchell]], [[Laraine Newman]] and [[Andrea Martin]].

1171 ¦ rowspan=&quot;6&quot; style=&quot;text-align:center;&quot;¦ [[21st Critics' Choice Awards¦2015]]&lt;br /&gt;&lt;ref&gt;{{cite web¦url=https://variety.com/2015/film/awards/critics-choice-award-nominations-2016-nominees-full-list-1201660235/¦title=Critics' Choice Award Nominations: Complete List¦last=Gray¦first=Tim¦date=December 14, 2015¦work=[[Variety (magazine)¦Variety]]¦access-date=December 14, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=https://variety.com/2016/film/news/critics-choice-awards-winners-list-2016-1201681953/¦title=Critics' Choice Awards: Winners List¦date=January 17, 2016¦work=[[Variety (magazine)¦Variety]]¦access-date=January 22, 2016}}&lt;/ref&gt; ¦¦ style=&quot;background:#B0C4DE;&quot; ¦ '''[[Sylvester Stallone]]''' ¦¦ style=&quot;background:#B0C4DE;&quot; ¦ '''[[Rocky Balboa]]''' ¦¦ style=&quot;background:#B0C4DE;&quot; ¦ '''''[[Creed (film)¦Creed]]'''''

1172 ''Leviathan'' was directed by [[George P. Cosmatos]], who had previously directed the [[Sylvester Stallone]]-starring action films ''[[Rambo: First Blood Part II]]'' and ''[[Cobra (1986 film)¦Cobra]]''.&lt;ref name=&quot;Knight 2008&quot; /&gt; The screenplay was written by [[David Peoples¦David Webb Peoples]] (''[[Blade Runner]]'') and [[Jeb Stuart (writer)¦Jeb Stuart]] (''[[Die Hard]]'').&lt;ref name=&quot;Knight 2008&quot; /&gt; [[Academy Award for Best Cinematography¦Oscar]]-nominated and [[British Society of Cinematographers¦BSC Award]]-winning cinematographer [[Alex Thomson (cinematographer)¦Alex Thomson]] served as the film's [[director of photography]]. Four-time [[Academy Award for Best Visual Effects¦Oscar]]-winning visual effects designer [[Stan Winston]] was responsible for the creature effects.&lt;ref name=&quot;Thomas 1989&quot; /&gt;

1173 Roop made a cameo appearance in the 1978 [[Sylvester Stallone]] movie ''[[Paradise Alley]]''.&lt;ref&gt;{{Cite web¦url=http://www.imdb.com/name/nm0740369/¦title=Bob Roop¦website=IMDb}}&lt;/ref&gt;

1174 The theatre was founded in 1946 with the current building's construction completed in 1953. The theatre is notable for having given birth to the acting careers of [[Steven Bauer]], [[Ray Liotta]], [[Sylvester Stallone]], and other University of Miami alumni. It is named in honor of 1953 University of Miami alumnus [[Jerry Herman]], whose [[Broadway theatre¦Broadway]] career included composing ''[[Hello, Dolly! (musical)¦Hello, Dolly!]]'', ''[[Mame (musical)¦Mame]]'', ''[[La Cage aux Folles (musical)¦La Cage aux Folles]]'', and other Broadway hits.

1175 *[[Sylvester Stallone]] - actor, ''[[Rocky (film series)¦Rocky]]'' and ''[[Rambo (film series)¦Rambo]]''

1176 &lt;s&gt;*'''Delete''' - this is already included in the [[Daniel Brandt]] article. We cannot justify this when we are questioning whether major supreme court cases should be included. This was just another draft dodger. We don't have articles about [[Sylvester Stallone]]'s draft dodger case, so why should we have one on Daniel Brandt's? Its utterly ridiculous. [[User:Zordrac¦Zordrac]] [[User_talk:Zordrac¦(talk)]] &lt;small&gt;[[M:AWWDMBJAWGCAWAIFDSPBATDMTD¦Wishy Washy]] [[M:Darwikinism¦Darwikinian]] [[M:Eventualism¦Eventualist]]&lt;/small&gt; 23:57, 15 December 2005 (UTC)&lt;/s&gt;

1177 *'''Keep'''. Where to start? First of all, I've never noticed any AfD about notability of a USSC case, all of which are of course notable. Even so, the USSC (and any other court) may often cite case law from other jursidictions, so there is certainly no reason that a case from a lower court should not be encyclopedic just because it's from a lower court. I can't tell much about how important this particularly case really has been, but a draft-era case in which someone you so pointedly refer to as a &quot;draft dodger&quot; was cleared by a federal circuit court seems unquestionably notable to me. If [[Sylvester Stallone¦Sly]] had a similar legal case at this level, it should certainly be mentioned in wikipedia (but I see no mention of anything like that). [[User:Bikeable¦Bikeable]] 01:25, 16 December 2005 (UTC)

1178 In 1982, Vajna and Kassar made their film production debut with the highly successful ''[[First Blood]]'', starring [[Sylvester Stallone]]. ''[[Rambo: First Blood Part II]]'' was released in 1985, generating more than $300 million worldwide, making it one of the most profitable films in the history of filmmaking.

1179 In 1995, Cinergi released ''[[Die Hard with a Vengeance]]'' starring [[Bruce Willis]], [[Jeremy Irons]], and [[Samuel L. Jackson]]. To date, the film has grossed over $300 million worldwide. That film was followed by ''[[Judge Dredd (film)¦Judge Dredd]]'' starring [[Sylvester Stallone]], ''[[The Scarlet Letter (1995 film)¦The Scarlet Letter]]'' with [[Demi Moore]], and [[Oliver Stone]]'s epic ''[[Nixon (film)¦Nixon]]'', starring [[Anthony Hopkins]]. The last film released was ''[[An Alan Smithee Film: Burn Hollywood Burn]]''.

1180 ¦ [[Sylvester Stallone]]

1181 ¦ [[Sylvester Stallone]] (O)

1182 * [[Sylvester Stallone]] is credited for writing and acting in ''[[Rocky]]''. [[Clint Eastwood]] is credited for directing and acting in ''[[Million Dollar Baby]]''. [[Warren Beatty]] and [[Woody Allen]] were each credited for acting, writing, and directing their respective movies on the list.

1183 ¦ [[Sylvester Stallone]], [[Kyra Sedgwick]], [[Thomas Jane]], [[Tom Berenger]], [[Cary Elwes]], [[Kelsey Grammer]], [[Danny Trejo]], [[Danny Aiello]], [[Tom Sizemore]]

1184 The film was lambasted by critics and is widely considered to be [[List of films considered the worst#The Garbage Pail Kids Movie (1987)¦one of the worst films ever made]]. ''The Garbage Pail Kids Movie'' was nominated for the [[Golden Raspberry Award for Worst New Star]] but lost to [[David Mendenhall]] for [[Sylvester Stallone]]'s ''[[Over the Top (1987 film)¦Over the Top]]'', and the song &quot;You Can Be a Garbage Pail Kid&quot; was nominated for the [[Razzie Award for Worst Original Song¦Worst Original Song]] but lost to [[George Michael]]'s controversial hit &quot;[[I Want Your Sex]]&quot; from [[Eddie Murphy]]'s ''[[Beverly Hills Cop II]]''.

1185 [[Image:Sylvester Stallone (1983).jpg¦thumb¦150px¦[[Sylvester Stallone]] in 1983]]

1186 In the 2012 movie ''[[The Expendables 2]]'', there is a wink at Chuck Norris facts when one is attributed to Norris' character Booker. In one scene Booker saves Barney Ross's ([[Sylvester Stallone]]) team in a firefight. After introductions all around, Ross says to Booker, &quot;I heard another rumor. That you were bitten by a [[king cobra]].&quot; &quot;Yeah, I was&quot;, Booker replies, adding, &quot;But after five days of agonizing pain, ''&lt;pause&gt;'' the cobra died.&quot;&lt;ref&gt;{{cite web¦ url=https://www.youtube.com/watch?v=feZB6eW5k_w&amp;t=164 ¦archive-url=https://ghostarchive.org/varchive/youtube/20211221/feZB6eW5k_w ¦archive-date=2021-12-21 ¦url-status=live¦ title= Chuck Norris scene in The Expendables 2 HD 720p¦ date=2012-11-26¦ publisher=OfficielFirat ¦ access-date=2019-08-29}}{{cbignore}}&lt;/ref&gt;

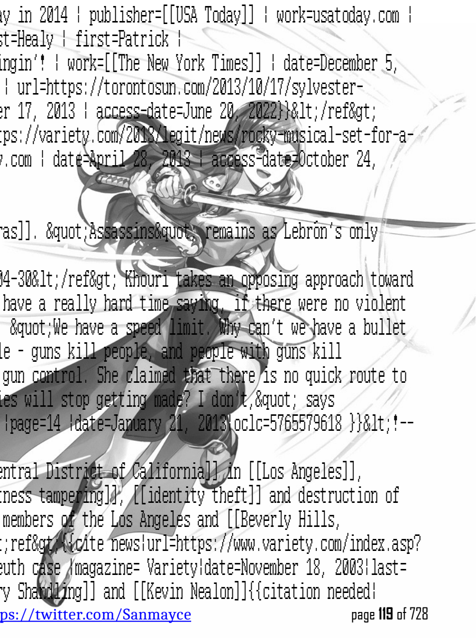
1187 In 2012, Meehan wrote the book from the original screenplay by [[Sylvester Stallone]]&lt;ref&gt;{{cite web ¦ last=Gardner ¦ first=Elysa ¦ url=https://www.usatoday.com/story/life/people/2013/04/28/rocky-musical-headed-to-broadway/2119697/ ¦ title='Rocky' musical bound for Broadway in 2014 ¦ publisher=[[USA Today]] ¦ work=usatoday.com ¦ date=April 28, 2013 ¦ access-date=December 27, 2013}}&lt;/ref&gt; for the musical [[Rocky the Musical¦''Rocky'']].&lt;ref&gt;{{cite web ¦ last=Healy ¦ first=Patrick ¦ url=https://www.nytimes.com/2012/12/09/theater/a-hit-in-germany-a-rocky-musical-aims-at-broadway.html?pagewanted=1 ¦ title=Yo, Adrian! I'm Singin'! ¦ work=[[The New York Times]] ¦ date=December 5, 2012 ¦ access-date=December 27, 2013}}&lt;/ref&gt; The show premiered in [[Hamburg]] in 2012,&lt;ref name=&quot;premiere&quot;&gt;{{cite web ¦ url=https://torontosun.com/2013/10/17/sylvester-stallone-talks-rocky-musical ¦ title=Sylvester Stallone talks 'Rocky' musical ¦ publisher=[[Toronto Sun]] ¦ work=torontosun.com ¦ date=October 17, 2013 ¦ access-date=June 20, 2022}}&lt;/ref&gt; before transferring to [[Broadway theatre¦Broadway]] in 2014.&lt;ref name=vault/&gt;&lt;ref&gt;{{cite web ¦ last=Cox ¦ first=Gordon ¦ url=https://variety.com/2013/legit/news/rocky-musical-set-for-a-bout-with-broadway-1200424086/ ¦ title='Rocky' Musical Set for a Bout With Broadway ¦ publisher=[[Variety (magazine)¦Variety]] ¦ work=variety.com ¦ date=April 28, 2013 ¦ access-date=October 24, 2013}}&lt;/ref&gt;

1188 * ''[[Rocky the Musical]]'' (with [[Sylvester Stallone]]) (2012)

1189 In 1995, Lebrón made his Hollywood debut in &quot;[[Assassins (1995 film)¦Assassins]]&quot;, with [[Sylvester Stallone]] and [[Antonio Banderas]]. &quot;Assassins&quot; remains as Lebrón's only Hollywood movie.

1190 According to an interview in ''Variety''&lt;ref&gt;[https://variety.com/t/callie-khouri/ Variety.com- Callie Khouri Profile] Retrieved 2015-04-30&lt;/ref&gt; Khouri takes an opposing approach toward guns in social media: &quot;in other countries where they have violent video games but less access to guns, they have less mass shootings. I have a really hard time saying, if there were no violent games, people would stop shooting each other. I think that until they have no way of shooting each other, they won't stop.&quot; says Khouri. &quot;We have a speed limit. Why can't we have a bullet limit? The idea that we don't need limits stricter than we have now on guns is absolutely insane. Because ultimately, people don't kill people - guns kill people, and people with guns kill people.&quot; She also argued that America has lost its moral compass and that &quot;it's worse than it's ever been!&quot; in the matters of gun control. She claimed that there is no quick route to end gun violence because Hollywood actors such as [[Sylvester Stallone]] &quot;make a fortune from violence. Do you think those types of movies will stop getting made? I don't,&quot; says Khouri.&lt;ref&gt;{{cite news¦url=https://www.worldcat.org/oclc/5765579618 ¦title=Voices: Callie Khouri ¦work=Variety ¦volume=429 ¦number=10 ¦page=14 ¦date=January 21, 2013¦oclc=5765579618 }}&lt;!--information from Worldcat record for the article, not the article itself, which seems to be unavailable online.--&gt;&lt;/ref&gt;

1191 On February 4, 2006, Pellicano was formally arrested on additional charges. On February 6, 2006, in [[United States District Court for the Central District of California]] in [[Los Angeles]], Pellicano was indicted on 110 counts. His alleged crimes included [[racketeering]], [[conspiracy (crime)¦conspiracy]], [[wiretapping]], [[witness tampering]], [[identity theft]] and destruction of evidence. Specifically, Pellicano was charged with receiving unlawfully accessed confidential records on celebrities and public figures from members of the Los Angeles and [[Beverly Hills, California¦Beverly Hills]] police departments. For example, prosecutors alleged that Pellicano tapped the phones of [[Sylvester Stallone]]&lt;ref&gt;{{cite news¦url=https://www.variety.com/index.asp?layout=story&amp;articleid=VR1117895933&amp;categoryid=22&amp;query=pellicano&amp;display=pellicano&amp;cs=1¦title=Stallone is queried in sleuth case ¦magazine= Variety¦date=November 18, 2003¦last=Shprintz ¦first= J.¦access-date=October 17, 2006}}&lt;/ref&gt; and [[Keith Carradine]] and accessed the confidential police records of [[Garry Shandling]] and [[Kevin Nealon]]{{citation needed¦



date=October 2021}}. On February 15, 2006, the indictment was amended to include further charges of wiretapping and extortion.&lt;ref&gt;{{cite news| url= https://news.yahoo.com/s/ap/20060206/ap_on_re_us/private_eye_indicted_1 |title= Private eye indicted| website= news.yahoo.com| agency= Associated Press| date= February 6, 2006| access-date= }}{{dead link|date=June 2016|bot=medic}}{{cbignore|bot=medic}}&lt;/ref&gt; Pellicano was denied bail. In June 2006 the ''[[Los Angeles Times]]'' reported that Pellicano performed an illegal [[background check]] on a law enforcement official who was investigating a fake passport scheme involving Pellicano's client, [[Christophe Rocancourt]].&lt;ref&gt;{{cite news|url=https://www.upi.com/Two-Hollywood-scandals-connect/33001149720289/|title=Two Hollywood scandals connect|publisher=United Press International|date=June 7, 2006| website= UPI.com |access-date=March 3, 2019}}&lt;/ref&gt;

1192 In the early years of Survivor, Peterik continued to co-write hits for other artists, including [[38 Special (band)|38 Special]] and [[Sammy Hagar]].&lt;ref name=&quot;LarkinHM&quot;&gt;{{cite book| title=[[Encyclopedia of Popular Music|The Guinness Who's Who of Heavy Metal]]|editor=Colin Larkin|editor-link=Colin Larkin (writer)|publisher=[[Guinness Publishing]]|date=1995|edition=Second|isbn=0-85112-656-1|page=274}}&lt;/ref&gt; In 1982, [[Sylvester Stallone]] commissioned Survivor to write and perform the theme song for ''[[Rocky III]]''. This song, &quot;[[Eye of the Tiger]]&quot;, became their defining single, spending six weeks at No. 1 on the [[Billboard Hot 100|''Billboard'' Hot 100]] and going double platinum. &quot;Eye of the Tiger&quot; also won a [[Grammy Award]] and resulted in an Oscar nomination for Peterik and Frankie Sullivan for Best Song.

1193 In 1978, he partnered with Jim Peterik and the two became a successful songwriting team.&lt;ref name=&quot;CP&quot; /&gt; With Sullivan as the lead guitarist, he and Peterik formed the nucleus of the band Survivor and the band began touring, playing concerts in clubs.&lt;ref name=&quot;Levine&quot;&gt;{{cite news |last1=Levine |first1=Gerald |title=Local rock band survives a hard climb to the top |url=https://newspaperarchive.com/other-articles-clipping-dec-30-1980-2759087/ |access-date=October 18, 2021 |publisher=Arlington Heights Daily Herald |date=December 30, 1980}}&lt;/ref&gt; Sullivan's first hit on the ''[[Billboard (magazine)|Billboard]]'' charts was in 1981 with the song &quot;Poor Man's Son&quot;, from the album ''[[Premonition (Survivor album)|Premonition]]''.&lt;ref name=&quot;CP&quot;&gt;{{cite news |title=Survivor to catch your eye with success |url=https://newspaperarchive.com/other-articles-clipping-jul-27-1993-2758987/ |access-date=October 18, 2021 | publisher=Clearfield Progress |date=July 27, 1993}}&lt;/ref&gt; Along with keyboardist and vocalist [[Jim Peterik]], Sullivan co-wrote all of the group's hits, including &quot;[[Eye of the Tiger]]&quot;&lt;ref&gt;{{cite web|title=Survivor 2012: A Conversation With Frankie Sullivan|url=http://technorati.com/entertainment/music/article/survivor-2012-a-conversation-with-frankie1/page-2/| publisher=Technorati|access-date=November 12, 2013|url-status=dead|archive-url=https://web.archive.org/web/20131112104340/http://technorati.com/entertainment/music/article/survivor-2012-a-conversation-with-frankie1/page-2/|archive-date=November 12, 2013}}&lt;/ref&gt; and &quot;[[Burning Heart (song)|Burning Heart]]&quot; from the ''[[Rocky III]]'' and ''[[Rocky IV|IV]]'' movie soundtracks.&lt;ref&gt;{{cite web|title=Frankie Sullivan|url=https://www.imdb.com/name/nm0838071/#soundtrack|publisher=IMDb|access-date=November 12, 2013}}&lt;/ref&gt; He credits [[Sylvester Stallone]] for giving him the &quot;opportunity of a lifetime&quot; by using his song &quot;Eye of the Tiger&quot; in the ''[[Rocky III]]'' movie.&lt;ref name=&quot;Republic&quot;&gt;{{cite news | last1=Blair |first1=Brian |title=A Survivor in rock world |url=http://www.therepublic.com/2019/08/22/a_survivor_in_rock_world/ |access-date=October 18, 2021 |publisher=The Republic |date=August 20, 2019}}&lt;/ref&gt;

1194 Survivor experienced their big breakthrough in 1982, when they were asked by [[Sylvester Stallone]] to provide the theme song for his movie ''[[Rocky III]]''. The song, &quot;[[Eye of the Tiger]]&quot;, soon turned into the biggest hit of the band's career. It had an enormous impact on the [[Billboard charts|''Billboard'' charts]], soaring to #1 for six weeks. The song won the band a [[Grammy award|Grammy Award]] for [[Grammy Award for Best Rock Performance by a Duo or Group with Vocal|Best Rock Performance By A Duo Or Group With Vocal]], was voted &quot;Best New Song&quot; by the [[People's Choice Awards]] and received an [[Academy Award]] nomination. The song remains popular today. It came in at #18 on the Top 100 Singles chart in ''Billboard''{{'}}s 100th Anniversary issue&lt;ref&gt;{{cite web|url=http://www.billboard.com/bbcom/specials/hot100/charts/top100-titles-30.shtml|archive-url=https://web.archive.org/web/20080913205956/http://www.billboard.com/bbcom/specials/hot100/charts/top100-titles-30.shtml|url-status=dead|archive-date=September 13, 2008|title=The Billboard Hot 100 All-Time Top Songs (30-21)|website=Billboard.com|accessdate=May 4, 2009}}&lt;/ref&gt; and it is well over the 300,000 mark on the [[iTunes Store]], where it once held #9 on their &quot;Top Soundtrack&quot; chart.{{citation needed|date=April 2020}} Thanks to the song's success and another Top 20 single, &quot;American Heartbeat&quot;, the album ''Eye of the Tiger'' peaked at #2 on the [[Billboard 200|''Billboard'' 200]] chart.

1195 Cannon worked on commercials for James Garrett and later RSA, and made his debut feature ''[[The Young Americans (film)|The Young Americans]]'' in 1993. His second film, the big-budget [[Sylvester Stallone]] feature ''[[Judge Dredd (film)|Judge Dredd]]'', released in 1995, was a commercial disappointment and was not well received critically. In 1998 he directed two features, the teen slasher film ''[[I Still Know What You Did Last Summer]]'' and he directed [[Ray Liotta]], [[Anjelica Huston]] and [[Jeremy Piven]] in ''[[Phoenix (1998 film)|Phoenix]]''.

1196 It was perhaps Z'Dar's performance in ''Maniac Cop'' that landed him the role of &quot;Face&quot; in 1989's ''[[Tango &amp; Cash]]'', alongside action stars [[Sylvester Stallone]] and [[Kurt Russell]].&lt;ref name=&quot;Lindenmuth2002&quot;&gt;{{cite book|last=Lindenmuth|first=Kevin J.|title=The independent film experience: interviews with directors and producers|url=https://books.google.com/books?id=xl6jii9wbzoC&amp;pg=PA80|access-date=September 20, 2011|date=January 2002|publisher=McFarland|isbn=978-0-7864-1075-0|page=80}}&lt;/ref&gt; He went on to appear in films such as ''Killing American Style'' (1988), ''[[The Final Sanction (film)|The Final Sanction]]'' (1990), ''[[A Gnome Named Gnorm]]'' (1990), ''[[Beastmaster 2: Through the Portal of Time]]'' (1991), ''[[Mobsters (film)|Mobsters]]'' (1991), ''[[Samurai Cop]]'' (1991), ''[[Return to Frogtown]]'' (1993), ''[[Marching Out of Time]]'' (1993), ''Death from Above'' (2011), ''Meltdown'' (2009) and ''Easter Sunday'' (2014).

1197 * [[Sylvester Stallone]]

1198 * [[Woody Allen]], [[Sharon Stone]], and [[Sylvester Stallone]] (''[[Antz]]'')

1199 In 2017 an annual film festival was established, the [[El Gouna Film Festival]].&lt;ref name=alarabiya&gt;{{cite web|url=http://english.alarabiya.net/en/life-style/entertainment/2017/09/23/ElGouna-Film-Festival-kicks-off-its-first-edition-in-Egypt.html|title=El Gouna Film Festival kicks off its first edition in Egypt|website=English.alarabiya.net|date=23 September 2017|access-date=7 October 2018}}&lt;/ref&gt; Founded by telecom billionaire [[Naguib Sawiris]],&lt;ref&gt;{{Cite web |last=Vivarelli |first=Nick |date=2023-02-22 |title=Egypt's El Gouna Film Festival Is Back After One Year Hiatus |url=https://variety.com/2023/film/global/egypt-el-gouna-film-festival-back-after-one-year-1235531694/ |access-date=2023-07-18 |website=Variety |language=en-US}}&lt;/ref&gt; El Gouna Film Festival was previously attended by globally prominent celebrities including ''[[Grey's Anatomy|Grey's Anatomy's]]'' [[Patrick Dempsey]],&lt;ref&gt;{{Cite web |date=2018-10-05 |title=Patrick Dempsey and Jean-Jacques Annaud present new TV series at the El Gouna Film Festival |url=https://www.euronews.com/2018/10/05/patrick-dempsey-and-jean-jacques-annaud-present-new-tv-series-at-the-el-gouna-film-fest |access-date=2023-07-18 |website=euronews |language=en}}&lt;/ref&gt; ''[[Rambo (2008 film)|Rambo's]]'' [[Sylvester Stallone]]&lt;ref&gt;{{Cite web |last=Vivarelli |first=Nick |date=2018-10-01 |title=Sylvester Stallone, 'A Land Imagined' Honored at El Gouna Film Festival |url=https://variety.com/2018/film/festivals/sylvester-stallone-el-gouna-singapore-a-land-imagined-wins-1202962868/ | access-date=2023-07-18 |website=Variety |language=en-US}}&lt;/ref&gt; and ''[[Bottle Rocket|Bottle Rocket's]]'' [[Owen Wilson]].&lt;ref&gt;{{Cite web |date=2018-10-05 |title=Owen Wilson describes toughest time of his career and reveals favourite film |url=https://www.euronews.com/2018/10/05/owen-wilson-describes-toughest-time-of-his-career-and-reveals-favourite-film |access-date=2023-07-18 | website=euronews |language=en}}&lt;/ref&gt;

1200 * In ''[[The Expendables (2010 film)|The Expendables]]'' (2010), numerous references are made to ravens and, obliquely, to &quot;The Raven&quot;. [[Sylvester Stallone]]'s character is in the process of getting a tattoo completed which features a raven, and the seaplane which his team travels in also features an oversize picture of a raven.

1201 In his last 20 years, Anderson remained one of the leading actors in Puerto Rico and continually acted in major local and international productions, including a small role as the bank director in the [[Sylvester Stallone]] thriller ''[[Assassins (1995 film)|Assassins]]''.

1202 ''No Exit'' received mixed to negative reviews. Writing for [[TV Guide]] and sister publication ''The Motion Picture Annual'', Robert Pardi found Wincott as limited as other action specialists such as [[Sylvester Stallone]], [[Dolph Lundgren]], [[Jean-Claude Van Damme]], and pointed to the purported hypocrisy of the film's anti-violence stance, saying: &quot;Since Wincott is a superb athlete, his fans may welcome this tedious trip to a death arena. But even diehard Wincott followers won't be fooled by the conspicuously shallow intercutting of flashbacks and flash-forwards, all leading to the same bloody conclusion: might makes right.&quot;&lt;ref&gt;{{cite book |editor=Grant, Edmond |last1=Pardi |first1=Robert |date=1998 |url=https://archive.org/details/motionpicturegui0000unse_d2w0/page/138/mode/2up |title=The Motion Picture - 1998 Annual (The Films of 1997) |location=New York |publisher=Cinebooks |pages=139-140 | isbn=0933997000}}&lt;/ref&gt; [[Ballantine Books]]' ''Video Movie Guide'' was not much impressed either, although it was more amicable towards its leading man, assessing that this &quot;[u]ltimately tedious ''[[Mortal Kombat]]''/''[[Street Fighter]]'' ripoff is buoyed somewhat by the undeniable video presence of star Jeff Wincott.&quot;&lt;ref&gt;{{cite book |last1=Martin |first1=Mick | last2=Porter |first2=Marsha |date=October 2001 |url=https://archive.org/details/isbn_9780345420961/page/364/mode/2up |title=Video Movie Guide 2002 |location=New York |publisher=Ballantine Books | page=365 |isbn=0345420969}}&lt;/ref&gt;

1203 They released only one album, ''Feel Lucky Punk?'', and an EP, ''Beginning to End'', on [[Mercury Records]] in 1995 and disbanded after the subsequent tour. &quot;Beginning to End&quot; was used in the soundtrack for the 1995 Sylvester Stallone film ''[[Assassins (1995 film)|Assassins]]''.&lt;ref&gt;{{Cite web|url=http://www.imdb.com/title/tt0112401/soundtrack|title=Assassins (1995) - IMDb| accessdate=22 September 2020|website=IMDb.com}}&lt;/ref&gt;

1204 * ''[[Nighthawks (1981 film)|Nighthawks]]'' (1981) - Wulfgar ([[Rutger Hauer]]) sets out to kill Irene DaSilva ([[Lindsay Wagner]]), unaware that it is Sgt. Dick DaSilva ([[Sylvester Stallone]]), who has donned women's clothing and a wig.

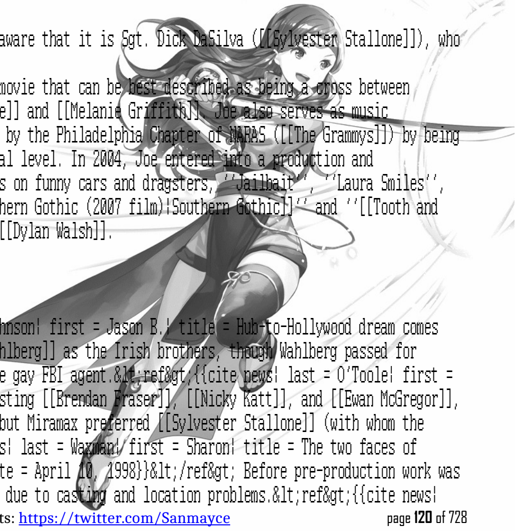
1205 In 2002, Joe joined forces with [[Merv Griffin]] and [[RKO Pictures]] as Executive Producer of the film [[Shade (film)|''Shade'']] - a movie that can be best described as being a cross between ''[[The Sting]]'' and ''[[Pulp Fiction]]''. The movie stars [[Jamie Foxx]], [[Gabriel Byrne]], [[Thandiwe Newton]], [[Sylvester Stallone]] and [[Melanie Griffith]]. Joe also serves as music supervisor on the film and will be overseeing the music on several other RKO films in the future. In April 2003, Joe Nicolo was honored by the Philadelphia Chapter of NARAS ([[The Grammys]]) by being awarded The Heroes Award as recognition of his outstanding work and contributions to the U.S. music industry on both a national and local level. In 2004, Joe entered into a production and distribution deal with Light-year/Warner Bros. Warners is currently distributing (for Joe's company Judgment Entertainment) a DVD series on funny cars and dragsters, ''Jailbait'', ''Laura Smiles'', as well as the Judgment Picture ''Shade''. Joe is also producing several new movie releases: ''[[Return to Sleepaway Camp]]'', ''[[Southern Gothic (2007 film)|Southern Gothic]]'' and ''[[Tooth and Nail (film)|Tooth and Nail]]''-both directed by Mark Young, and ''[[One Part Sugar]]'' starring [[Danny DeVito]], [[Justin Long]], and [[Dylan Walsh]].

1206 [[Category:Images of Sylvester Stallone]]</text>

1207 | 1985 || ''[[Rambo: First Blood Part II]]'' || [[Georges P. Cosmatos]] || With [[Sylvester Stallone]]

1208 * [[Demolition Man (film)|''Demolition Man'' (film)]], a 1993 film starring Sylvester Stallone and Wesley Snipes

1209 Filming of ''The Boondock Saints'' was scheduled for the coming autumn in Boston.&lt;ref name=&quot;hub&quot;&gt;{{cite news| last = Johnson| first = Jason B.| title = Hub-to-Hollywood dream comes true for scriptwriter| work = [[Boston Globe]]| date = April 1, 1997}}&lt;/ref&gt; Duffy sought to cast [[Stephen Dorff]] and [[Mark Wahlberg]] as the Irish brothers, though Wahlberg passed for ''[[Boogie Nights]]''. The director also wanted to cast [[Billy Connolly]] and [[Kenneth Branagh]] in the film, with Branagh playing the gay FBI agent.&lt;ref&gt;{{cite news| last = O'Toole| first = Lesley| title = Hollywood's young guns| work = [[The Times]]| date = December 4, 1997}}&lt;/ref&gt; Duffy also expressed interest in casting [[Brendan Fraser]], [[Nicky Katt]], and [[Ewan McGregor]], with two of them as the brothers, but no decisions were finalized. The director later sought [[Patrick Swayze]] to play the FBI agent, but Miramax preferred [[Sylvester Stallone]] (with whom the studio had an existing relationship), [[Bill Murray]] or [[Mike Myers (actor)|Mike Myers]].&lt;ref name=&quot;faces&quot;&gt;{{cite news| last = Waxman| first = Sharon| title = The two faces of Hollywood: A screenwriter's success story had a happy ending. Then someone changed the script.| newspaper = [[The Washington Post]]| date = April 10, 1998}}&lt;/ref&gt; Before pre-production work was supposed to begin in Boston in December 1997, Miramax pulled out of the project. Producer Lloyd Segan said that the project had stalled due to casting and location problems.&lt;ref&gt;{{cite news|



last = Raposa| first = Laura| title = Miramax 'Saints' hits devil of a snag| work = [[Boston Herald]]| date = December 13, 1997}}&lt;/ref&gt; While Duffy was able to keep the writer's fee of $300,000, the studio required the reimbursement of the $150,000 director's fee and the $700,000 advance to develop the project.

1210 In 1988, McAlary wrote a non-fiction book, ''Buddy Boys'', about corrupt police in New York's 77th Precinct, in the Brooklyn North patrol borough.&lt;!--source is our article--&gt; He also had a hand in writing the script for the movie ''[[Cop Land]]'', starring [[Sylvester Stallone]] and [[Robert De Niro]].

1211 In 2015, [[Jim Harbaugh]] was quoted saying, &quot;I've been a Brut man since I was 10 years old.&quot;&lt;ref&gt;{{Cite news|url=https://www.freep.com/story/sports/columnists/jeff-seidel/2015/05/18/jim-harbaugh-entertaining-far-michigan-nothings-done-yet/27566049/|title=Jeff Seidel: Harbaugh entertaining so far at U-M, but nothing's done yet|first=Jeff|last=Seidel|newspaper=[[Detroit Free Press]]}}&lt;/ref&gt; [[Elvis Presley]] used Brut.&lt;ref&gt;{{Cite web|url=https://celebrityfragranceguide.com/what-cologne-does-he-wear/elvis-presley|title=Elvis Presley|website=Celebrityfragranceguide.com|date=October 10, 2016}}&lt;/ref&gt; In the film ''[[Over the Top (1987 film)|Over the Top]]'', [[Sylvester Stallone]] picks up a [[tractor trailer]] of Brut in California before taking his son on an eastbound cross-country trip.

1212 After &quot;Rocky VI (AKA Rocky Balboa)&quot;, will Sylvester Stallone make a 7th film in the series?

1213 [[Sylvester Stallone|Sly]] would be best advised to stop making them before he looks like Rocky [http://www.tv-nostalgie.de/james%20garner%20und%20noah%20beery.jpg] from [[The Rockford Files]], to avoid any potential confusion. [[User:StuRat|StuRat]] 18:05, 4 February 2006 (UTC)

1214 * ''[[Cop Land]]'', a 1997 movie starring Sylvester Stallone, Robert De Niro and Harvey Keitel

1215 *[[Sylvester Stallone]]

1216 | Main role&lt;ref&gt;{{cite web |last1=White |first1=Peter |date=May 6, 2022 |title=''Tulsa King'': Andrea Savage To Star Alongside Sylvester Stallone In Taylor Sheridan's Paramount+ Mob Drama Series |url=https://deadline.com/2022/05/tulsa-king-andrea-savage-sylvester-stallone-taylor-sheridan-1235015650/ |website=[[Deadline Hollywood]] |publisher=[[Penske Media Corporation]]}}&lt;/ref&gt;

1217 In 1998, de la Fuente was contracted by [[Sony]] and [[Telemundo]] for a drama series remake of ''[[Starsky and Hutch]]'' called ''Reyes y Rey''. The weekly hour format did not catch on with Hispanic viewers and it was soon cancelled. He was a guest star on ''[[Pensacola: Wings of Gold]]''. He has also appeared regularly on ''[[CSI: Miami]]'' and ''[[Family Law (American TV series)|Family Law]]''. He has starred opposite [[Sylvester Stallone]] in ''[[Driven (2001 film)|Driven]]'', in [[John Carpenter]]'s ''[[Vampires: Los Muertos]]'' with [[Jon Bon Jovi]], as well as the independent thriller ''Minimal Knowledge''.

1218 Solar Films was then formed in 1988 (one of the first films that distributed by Solar Films is [[Rambo III]] of [[Sylvester Stallone]]), bringing international films (and later, local films since 2009) to the Philippines. Since 2012, Solar has collaborated with independent film director [[Brillante Mendoza]] for a multiple film contract.&lt;ref&gt;{{Cite web |url=http://www.solarentertainmentcorp.com/about/page/solar_today |title=Solar Today |access-date=May 15, 2014 |archive-url=https://web.archive.org/web/20140517114818/http://www.solarentertainmentcorp.com/about/page/solar_today |archive-date=May 17, 2014 |url-status=dead }}&lt;/ref&gt; On March 18, 2015, Solar launched the Sinag Maynila film festival is, which featured five films from a talented group of directors, each with a different vision and approach. The films were shown in select SM Cinemas from March 18 to 24, 2015. The winners were announced at the "Gabi ng Parangal" on March 22.

1219 * [[Sylvester Stallone]]

1220 McPhee released a jewelry line, KMF Jewelry, in early November 2022.&lt;ref&gt;{{cite news|url=https://people.com/celebrity/star-tracks-saturday-november-12-2022/|title=Mindy Kaling Gets Her Game On in L.A., Plus Sylvester Stallone, Katharine McPhee and More|work=People|date=November 12, 2022|access-date=November 12, 2022}}&lt;/ref&gt;

1221 In 2008, The Arches reprieved its ''Alien War'' attraction, which originally took place in 1992, and transformed it into ''Alien Wars''. The initial show was an [[Alien (franchise)|Alien]]-inspired tour, recreating the atmosphere and horror of the Alien films. Many celebrities were linked to this attraction, when staged in London, including: [[Sigourney Weaver]], who participated; [[Sylvester Stallone]], whose restaurant was next door, and declined to take part; and, [[Michael Jackson]],&lt;ref name=&quot;BBCAW&quot;&gt;{{cite web |url=http://www.bbc.co.uk/blogs/thereporters/paulinemclean/2008/12/alien_war.html |title=Alien War |access-date=2020-11-25 | last=McLean | first=Pauline |date=2008-12-05 |publisher=[[BBC]]}}&lt;/ref&gt; whose bodyguards visited but apparently didn't finish the tour.&lt;ref name=&quot;thelistAW&quot;&gt;{{cite web |url=http://www.list.co.uk/article/14594-alien-war/ |title=Alien War |access-date=2020-11-25 |last=Northmore |first=Henry |date=2008-11-27 | work=[[The List (magazine)|The List]]}}&lt;/ref&gt; Sigourney Weaver, when asked to comment on her experience, said &quot;Although I have been through the movies, I was screaming as much as everyone else.&quot;&lt;ref name=&quot;thelistAW&quot; /&gt;

1222 *The 1982 movie [[First Blood]] (starring [[Sylvester Stallone]] as [[John Rambo|Rambo]]) was shot in Golden Ears Provincial Park.

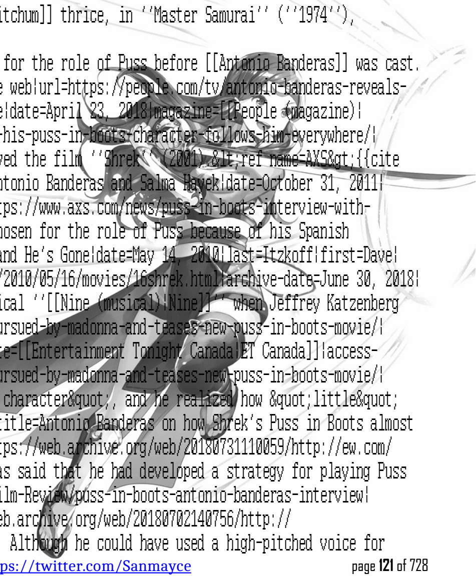
1223 Gardell is known for his comedic roles and is influenced by [[Jackie Gleason]].&lt;ref name=&quot;USAToday-2011&quot;/&gt; He has also listed [[Richard Pryor]], [[George Carlin]], [[John Candy]],&lt;ref&gt;{{cite web|last=Duck|first=Allison|url=https://lasvegasweekly.com/ae/2011/jul/06/comedian-billy-gardell-hits-halftime/|title=Comedian Billy Gardell hits 'Halftime'|date=July 6, 2011|work=[[Las Vegas Weekly]]|access-date=February 3, 2013}}&lt;/ref&gt; and [[John Belushi]] as his influences.&lt;ref&gt;{{cite web|last=Oldenburg|first=Ann|url=http://www.usatoday.com/life/people/story/2011-10-02/Billy-Gardell-Mike-amp-Molly-Melissa-McCarthy/50637080/1|title='Mike &amp; Molly' star Billy Gardell is one happy husband|date=November 3, 2011|work=[[USA Today]]|access-date=February 3, 2013|archive-url=https://web.archive.org/web/20120107150844/http://www.usatoday.com/life/people/story/2011-10-02/Billy-Gardell-Mike-amp-Molly-Melissa-McCarthy/50637080/1|archive-date=January 7, 2012|url-status=dead}}&lt;/ref&gt; Although Bonkerz helped launch his comedy career, he has credited the support from his grandmother, saying: &quot;She told me when I was 8 years old that I could be a comic&quot;.&lt;ref name=&quot;OrlandoSentinel-2001&quot;/&gt; Before ''Mike &amp; Molly'', Gardell revealed that he considered a career in radio after being on the road for his comedy act began to affect his family time.&lt;ref name=&quot;USAToday-2011&quot;/&gt; He has performed in several feature films, including ''[[Bad Santa]]'' (with [[Billy Bob Thornton]]) ''[[You, Me, and Dupree]]'' (with [[Owen Wilson]]) and ''[[Avenging Angelo]]'' (with [[Sylvester Stallone]] and [[Anthony Quinn]]) as well as appearing numerous times in recurring roles in several US television series, including [[NBC]]'s ''[[Heist (2006 TV series)|Heist]]'', ''[[The Practice]]'', ''[[Yes, Dear]]'', ''[[Desperate Housewives]]'', ''[[Lucky (American TV series)|Lucky]]'', ''[[Bones (TV series)|''Bones'']]'', ''[[My Name Is Earl]]'', ''[[Monk (TV series)|Monk]]'', and ''[[The King of Queens]]''. Gardell appeared as himself on the [[Comedy Central]] series ''[[Make Me Laugh]]''. Subsequently, he appeared on Miller's television program ''Dennis Miller'', telecast on the US financial news network [[CNBC]] (conducting &quot;man-on-the-street&quot; interviews). Notably, he created and staged an ''[[Saturday Night Live|SNL]]''-style charity performance of a piece called ''Winter Park Live'', the proceeds from which were donated to [[Comic Relief (charity)|Comic Relief]].

1224 Viktor Manoel has also appeared in film such as ''[[Staying Alive (1983 film)|Staying Alive]]''&lt;ref name=NYT/&gt; (1983) directed by [[Sylvester Stallone]], ''[[Breakin' 2: Electric Boogaloo]]''&lt;ref name=NYT/&gt; (1984), ''[[Glass Spider]]'' (1988) directed by David Mallet, and ''Female Perversions'' (1997) directed by Susan Streitfeld.

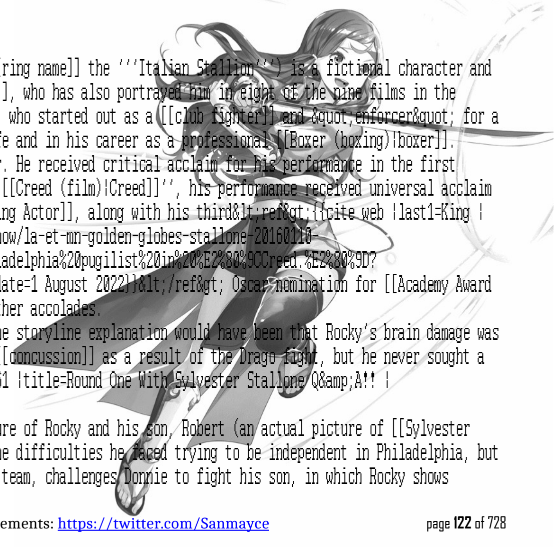
1225 In May 2022, Ayer signed on to direct action thriller ''[[The Beekeeper (2024 film)|The Beekeeper]]'' for [[Miramax]], written by [[Kurt Wimmer]] and starring [[Jason Statham]], with [[Metro-Goldwyn-Mayer]] later acquiring domestic distribution rights.&lt;ref&gt;{{Cite web|last=Wiseman|first=Andreas|date=May 5, 2022|title='Suicide Squad' Filmmaker David Ayer To Direct Jason Statham In Miramax's Action Pic 'The Beekeeper' – Cannes Market Hot Package|url=https://deadline.com/2022/05/jason-statham-david-ayer-suicide-squad-beekeeper-movie-miramax-cannes-1235017464/|access-date=November 18, 2022|website=[[Deadline Hollywood]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|last=Wiseman|first=Andreas|date=August 31, 2022|title=Jason Statham-David Ayer Action Pic 'The Beekeeper' Pre-Bought By MGM For U.S. &amp; Some Of International|url=https://deadline.com/2022/08/suicide-squad-david-ayer-jason-statham-beekeeper-mgm-miramax-1235104064/|access-date=November 18, 2022|website=[[Deadline Hollywood]]}}&lt;/ref&gt; In October 2023, it was announced that Ayer and Statham would reunite to work on ''Levon's Trade'', from a screenplay by [[Sylvester Stallone]], based on the novel of the same name by [[Chuck Dixon]], with production expected to begin in March 2024.&lt;ref&gt;{{Cite web|last=Wiseman|first=Andreas|date=October 27, 2023|title=Jason Statham To Star In David Ayer-Directed Sylvester Stallone Script 'Levon's Trade' From Black Bear &amp; Bill Block's BlockFilm – AFM Hot Package|url=https://deadline.com/2023/10/jason-statham-david-ayer-sylvester-stallone-levons-trade-black-bear-bill-block-afm-1235585741/|access-date=January 11, 2024|website=[[Deadline Hollywood]]}}&lt;/ref&gt;

1226 Gale directed [[Richard Harrison (actor)|Richard Harrison]] in ''Intrusion Cambodia'' AKA ''Rescue Team'' (1981), also starring a cast of ''Silver Star'' workhorses such as [[Romano Kristoff]], James Gaines, and [[Mike Monty]]. The [[assistant director]] of ''Intrusion Cambodia'' was [[Teddy Page]], another prolific ''Silver Star'' director. Gale directed Kristoff and Monty again in ''Slash'' (1984), a film influenced by the [[Rambo (film series)|Rambo]] series starring [[Sylvester Stallone]], to the point of plagiarism. ''Slash'' also featured [[Nick Nicholson (actor)|Nick Nicholson]] and Gwendolyn Hung. Another film of debatable note was ''Commando Invasion'' (1986), which featured [[Gordon Mitchell]], Ken Watanabe (not to be confused with the more famous [[Ken Watanabe (actor)|Ken Watanabe]]) and former model [[Tetchie Agbayani]] (as Carol Roberts, also in ''Intrusion Cambodia''). Gale also directed [[Christopher Mitchum]] thrice, in ''Master Samurai'' (''1974''), ''Commander Firefox'' (1983) and ''SFX Retaliator'' (1987).

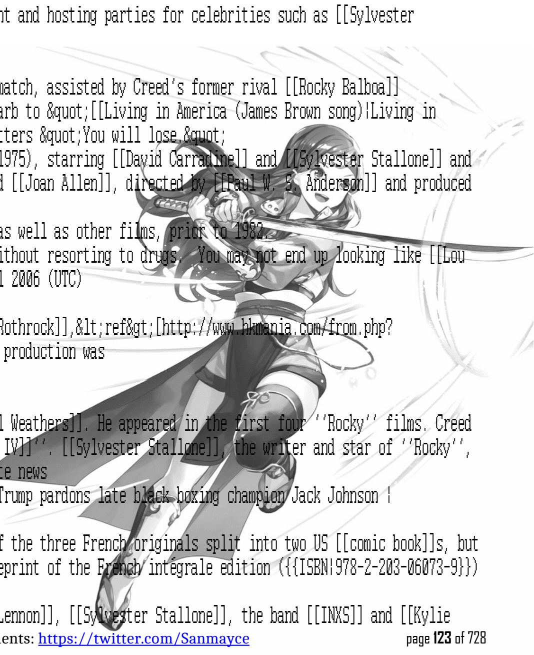
1227 [[Robert Patrick]], [[Harrison Ford]], [[Sylvester Stallone]], [[David Krumholtz]], [[Dennis Quaid]], and [[Todd Stashwick]] were considered for the role of Puss before [[Antonio Banderas]] was cast. Antonio Banderas voiced Puss in the ''Shrek'' franchise.&lt;ref name=AnimationWorldNetwork/&gt;&lt;ref name=TheAtlantic/&gt;&lt;ref&gt;{{cite web|url=https://people.com/tv/antonio-banderas-reveals-how-his-puss-in-boots-character-follows-him-everywhere/|title=Antonio Banderas Reveals How His Puss in Boots Character Follows Him Everywhere|date=April 23, 2018|magazine=[[People (magazine)|People]]|access-date=July 31, 2018|archive-url=https://web.archive.org/web/20180731125722/https://people.com/tv/antonio-banderas-reveals-how-his-puss-in-boots-character-follows-him-everywhere/|archive-date=July 31, 2018|url-status=live|df=mdy-all}}&lt;/ref&gt; Banderas said that his initial motivation to voice Puss was that he enjoyed the film ''Shrek'' (2001).&lt;ref name=AXS&gt;{{cite web|url=https://www.axs.com/news/puss-in-boots-interview-with-antonio-banderas-and-salma-hayek-100107|title='Puss in Boots' interview with Antonio Banderas and Salma Hayek|date=October 31, 2011|last=Lebowitz|first=Steven|website=[[AXS (company)|AXS]]|access-date=June 30, 2018|archive-url=https://web.archive.org/web/20180630141040/https://www.axs.com/news/puss-in-boots-interview-with-antonio-banderas-and-salma-hayek-100107|archive-date=June 30, 2018|url-status=dead|df=mdy-all}}&lt;/ref&gt; According to the actor, he was chosen for the role of Puss because of his Spanish accent.&lt;ref name=TheNewYorkTimes&gt;{{cite web|url=https://mobile.nytimes.com/2010/05/16/movies/16shrek.html|title=He's Big, He's Green, and He's Gone|date=May 14, 2010|last=Itzkoff|first=Dave|newspaper=[[The New York Times]]|access-date=June 30, 2018|archive-url=https://web.archive.org/web/20180630140537/https://mobile.nytimes.com/2010/05/16/movies/16shrek.html|archive-date=June 30, 2018|url-status=dead|df=mdy-all|author-link=Dave Itzkoff}}&lt;/ref&gt; According to Banderas, he was on [[Broadway theatre|Broadway]] for the musical ''[[Nine (musical)|Nine]]'' when Jeffrey Katzenberg approached him about taking the role.&lt;ref name=ETCanada&gt;{{cite web|url=https://etcanada.com/news/532979/antonio-banderas-talks-being-pursued-by-madonna-and-teases-new-puss-in-boots-movie/|title=Antonio Banderas Talks Being Pursued By Madonna And Teases New 'Puss In Boots' Movie|date=October 31, 2019|last=Atad|first=Corey|website=[[Entertainment Tonight Canada|ET Canada]]|access-date=November 7, 2019|archive-url=https://web.archive.org/web/20191107163313/https://etcanada.com/news/532979/antonio-banderas-talks-being-pursued-by-madonna-and-teases-new-puss-in-boots-movie/|archive-date=November 7, 2019|url-status=dead|df=mdy-all}}&lt;/ref&gt; The ''Shrek 2'' filmmakers showed him &quot;a lot of paintings of the character&quot;, and he realized how &quot;little&quot; Puss was.&lt;ref name=EntertainmentWeekly&gt;{{cite web|url=http://ew.com/movies/2018/06/07/puss-in-boots-different-voice-antonio-banderas/|title=Antonio Banderas on how Shrek's Puss in Boots almost sounded different|date=June 7, 2018|last=Lenker|first=Maureen Lee|magazine=[[Entertainment Weekly]]|access-date=July 31, 2018|archive-url=https://web.archive.org/web/20180731110059/http://ew.com/movies/2018/06/07/puss-in-boots-different-voice-antonio-banderas/|archive-date=July 31, 2018|url-status=dead|df=mdy-all}}&lt;/ref&gt; Banderas said that he had developed a strategy for playing Puss after accepting the role, which had determined Puss's personality.&lt;ref name=IndieLondon2&gt;{{cite web|url=http://www.indielondon.co.uk/Film-Review/puss-in-boots-antonio-banderas-interview|title=Puss in Boots - Antonio Banderas interview|last=Carnevale|first=Rob|website=IndieLondon|access-date=July 2, 2018|archive-url=https://web.archive.org/web/20180702140756/http://www.indielondon.co.uk/Film-Review/puss-in-boots-antonio-banderas-interview|archive-date=July 2, 2018|url-status=dead|df=mdy-all}}&lt;/ref&gt; Although he could have used a high-pitched voice for



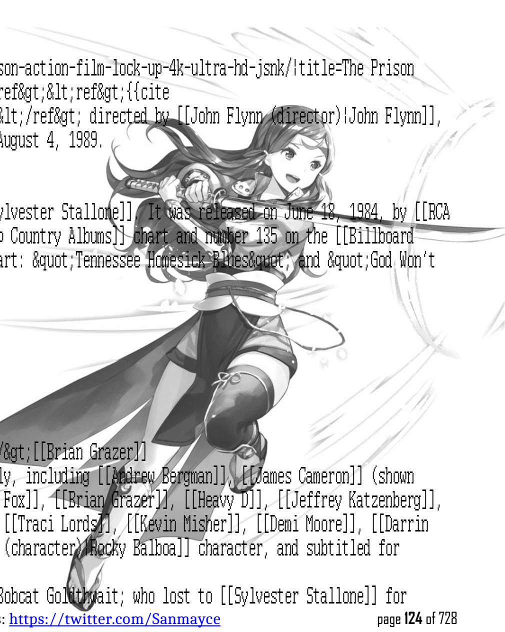
```
Puss, which was the filmmakers' original idea,&lt;ref name=ETCanada/&gt; he and the others working on the film opted for a tone that was &quot;deeper&quot; and &quot;more breathy&quot; than his normal voice.&lt;ref name=Collider2/&gt;&lt;ref name=IndieLondon2/&gt; Banderas called the choice &quot;very interesting&quot;, adding that it &quot;helped to establish the limits and the parameters of the character in terms of personality&quot;.&lt;ref name=IndieLondon2/&gt; He regarded the effect as &quot;almost like a lion trapped in the body of a little cat&quot;, which makes Puss &quot;different&quot;.&lt;ref name=IndieLondon2/&gt; According to the actor, Puss's voice contrasts with his body; he stated that &quot;the cat is not supposed to talk like that&quot;, adding that the difference between Puss's voice and appearance is [[comic relief]].&lt;ref name=Collider2/&gt; He also commented that the contrast between Puss's appearance and voice makes it seem like he is not even aware of his size.&lt;ref name=ETCanada/&gt; Banderas said that after the decision about Puss's voice had been made, the filmmakers had begun depicting the character &quot;in a totally different way&quot;.&lt;ref name=IndieLondon2/&gt; He added that Puss had initially been conceptualized as &quot;quite a little character&quot; but had started gaining more importance after the filmmakers had realized his potential.&lt;ref name=IndieLondon2/&gt; Banderas said that he and the filmmakers had &quot;a lot of fun&quot; with Puss's character, and he felt that viewers did too.&lt;ref name=Collider3/&gt;
1228 ''Lionheart'' has its roots in two different projects. Van Damme had written an outline for an underground fighting film, called ''The Wrong Bet''. Meanwhile, his friend Sheldon Lettich was working on a script about the French Foreign Legion intended for [[Sylvester Stallone]], for whom he had already written ''[[Rambo III]]''. Van Damme occasionally helped Lettich with [[French language]] terms. When that film did not get made, it was decided to integrate the legionnaire backstory into ''The Wrong Bet'' to flesh out its central character.&lt;ref name=&quot;lettich-bio&quot;/&gt; A significant inspiration for the film was 1975's ''[[Hard Times (1975 film)|Hard Times]]'', a favorite of both Van Damme and Lettich. ''The Wrong Bet'' followed a similar template, pairing a somber hero with a shifty yet endearing manager (respectively played by [[Charles Bronson]] and [[James Coburn]] in the classic film).&lt;ref name=&quot;lettich-bio&quot;/&gt;
1229 Among the notable residents are hockey great [[Wayne Gretzky]], who lived in a Georgian-style estate designed by fellow Canadian, Richard Landry. Gretzky sold the estate in 2007 for 18.5 million to another sports-star, baseball's [[Lenny Dykstra]] and moved to a slightly less extravagant mansion in the community. Since then, Dykstra's financial and legal problems forced him to sell the estate to a private individual and move away from the community. Dykstra declared bankruptcy, causing him to lose the property to foreclosure and in 2018 Gretzky repurchased the same property that he previously sold to Dykstra.&lt;ref&gt;{{cite web|url=http://www.latimes.com/business/realestate/hot-property/la-fi-hp-hotprop-20180505-story.html|title=Wayne Gretzky shells out $13.5 million for a familiar place in Thousand Oaks|first=Neal J. Leitereg, Jack|last=Flemming|website=latimes.com}}&lt;/ref&gt; Singer [[Paul Anka]] moved to Lake Sherwood in 2007 and owns multiple properties in the exclusive neighborhood. [[Tom Petty]] lived in a lakefront retreat that was originally built in 1931. Producer/Author/Journalist Larry Garrison resides with his wife. The tennis legend [[Pete Sampras]] and his wife, actress [[Bridgette Wilson]], built a very large, modern, hilltop estate they have since moved from the house. [[Britney Spears]] moved to the community in 2011 but recently moved into nearby Hidden Valley. Many other famous inhabitants own property in the community, however, they are only part-time residents and use their estates as a second home or as a retreat. [[Sylvester Stallone]] owned a lakefront hideaway that was featured in the film [[American Pie (film)|American Pie]]. Also, [[William Randolph Hearst]], who was one of the first famous residents in the area and owned property around the lake, used his estate as a vacation getaway during the 1920s and 1930s.
1230 His second feature-length film, ''[[Twenty Four Seven (film)|Twenty Four Seven]]'', won several awards at film festivals, including the [[Douglas Hickox]] award at the [[British Independent Film Awards]] and Best Screenplay at the [[Thessaloniki Film Festival]]. ''[[Dead Man's Shoes (2004 film)|Dead Man's Shoes]]'', his sixth film, and third starring [[Paddy Considine]],&lt;ref name=&quot;Tel061004&quot;&gt;Blacklock, Mark; [https://www.telegraph.co.uk/culture/film/3625051/Cruel-justice.html &quot;Cruel justice&quot;] ''Telegraph.co.uk'', 6 October 2004&lt;/ref&gt; was nominated for a [[British Academy of Film and Television Arts|BAFTA]] for Best British Film. His seventh film ''[[This is England]]'', won the [[British Independent Film Awards 2006]] for best British independent film. Meadows was presented with the award by [[Sylvester Stallone]] and used the occasion to announce that he was to be a father. ''[[This is England]]'' also won a [[British Academy of Film and Television Arts|BAFTA]] for Best British Film.
1231 *It is featured in the 1995 action movie ''[[Judge Dredd (film)|Judge Dredd]]'', starring [[Sylvester Stallone]].
1232 * ''[[Rambo: First Blood Part II]]'' (1985-US): [[Sylvester Stallone]]'s super soldier goes to Vietnam (actually Thailand) looking for his [[POW]] buddies. Followed by ''[[Rambo III]]'', set in [[Afghanistan]], but partially shot in Thailand.
1233 * ''[[American Gangster (film)|American Gangster]]'' (2007-US): Directed by [[Ridley Scott]] and starring [[Denzel Washington]] and [[Russell Crowe]], the story of an American [[heroin]] [[Illegal drug trade|smuggler]] was filmed in November 2006 in [[Chiang Mai Province|Chiang Mai]].&lt;ref&gt;Thepararat, Chatrudee. October 26, 2006. [http://www.bangkokpost.com/Business/26Oct2006_biz35.php &quot;Denzel and Sly head for blockbuster shoots in Thailand&quot;], ''Bangkok Post''.&lt;/ref&gt;&lt;ref&gt;Agence France Press. 26 October 2006. [http://nationmultimedia.com/2006/10/25/headlines/headlines_30017109.php Denzel Washington, Sylvester Stallone to shoot films in Thailand], via ''The Nation''.&lt;/ref&gt;
1234 * ''[[Rambo (2008 film)|Rambo]]'' (2007-US): [[Sylvester Stallone]] returned to Thailand to make the fourth installment in his ''[[Rambo (film series)|Rambo]]'' franchise, directing and starring as the Vietnam War veteran who takes on a mission to protect Christian [[missionary|missionaries]] delivering aid to the [[Karen people]] in [[Myanmar]]. Filming was due to start in January 2007.&lt;ref&gt;Deutsche Presse-Agentur. [http://www.bangkokpost.com/breaking_news/breakingnews.php?id=113484 &quot;Rambo to rescue Karen - in the movies&quot;], via ''Bangkok Post'' (retrieved October 11, 2006).&lt;/ref&gt;&lt;ref&gt;Production Weekly. October 6, 2006. [http://productionweekly.com/2006/10/06/stallone-looks-in-the-serpents-eye/ &quot;Stallone looks 'In the Serpent's Eye'&quot;] {{webarchive|url=https://web.archive.org/web/20061105215916/http://productionweekly.com/2006/10/06/stallone-looks-in-the-serpents-eye/ |date=2006-11-05 }} (retrieved October 11, 2006).&lt;/ref&gt;
1235 In 2013, Winter wrote the screenplay to [[Martin Scorsese]]'s ''[[The Wolf of Wall Street (2013 film)|The Wolf of Wall Street]]'' for which he was nominated for an [[Academy Award for Best Adapted Screenplay]]. He was also the co-creator, writer and executive producer of another HBO television drama series, ''[[Vinyl (TV series)|Vinyl]]'' (2016), which ran for one season. He was the co-showrunner on the [[Paramount+]] crime series ''[[Tulsa King]]'' with [[Taylor Sheridan]].&lt;ref&gt;{{cite web |last1=White |first1=Peter |title=Sylvester Stallone Set To Star In Taylor Sheridan &amp; Terence Winter Drama Series 'Kansas City' For Paramount+ From 101 Studios &amp; MTV Ent. Studios |url=https://deadline.com/2021/12/sylvester-stallone-taylor-sheridan-terence-winter-drama-series-kansas-city-paramount-1234883070/ |website=Deadline Hollywood |publisher=Penske Media Corporation |access-date=December 12, 2021 |language=English |date=December 6, 2021 }}&lt;/ref&gt;
1236 * '''Assy McGee''' ([[Larry Murphy (actor)|Larry Murphy]]) - a walking, talking lower torso with exposed [[buttocks]]. Horribly violent, an alcoholic and clinically depressed, Assy is a parody of the 1970s/1980s movie cops as seen in such films as ''[[Dirty Harry]]'', ''[[Lethal Weapon]]'' and ''[[Cobra (1986 film)|Cobra]]'': trigger-happy, tough, at times hopelessly depressed, and in conflict with his fellow officers as often as he is in conflict with crime.  Assy has a slurred style of speech similar to [[Sylvester Stallone]], though it is &quot;gassy&quot; sounding and muffled, possibly due to his anatomy or inebriation (or both), however has been shown to have an excellent ability to sing classical music.  He also quotes and mimics some of Stallone's most memorable characters, shouting Judge Dredd's catch phrase &quot;I am the law!&quot; in episode five &quot;Mexican Rain&quot; and reenacts the Rocky training scene in episode seven &quot;Ring of Fire&quot;.  Sometimes, when out of breath, or even just peeved, Assy will [[flatulate]]. Assy walks backwards and even jumps backwards. Assy has stated that he is of [[Cuba]]n descent. His Cuban heritage is backed up in the episode, &quot;Conviction&quot; when a childhood picture of Assy with a Cuban hat is shown in the background. We learn in the episode &quot;Hands Up&quot; (from season two) that Assy served in Vietnam, when a war flashback causes him to accidentally fire a bullet from an AK-47 into the abdomen of a World War II veteran. It is unclear whether Assy possesses [[genitalia]], though it is strongly implied on a few occasions. He has been shown walking in profile with no visible genital protrusion or covering, but he requested a &quot;happy ending&quot; from a [[masseuse]] in the episodes &quot;The Flirty Black Man&quot;, and &quot;Murder By The Docks&quot;. He is also perfectly capable of urinating in the [[Western canon|canon]]ical fashion. Licensed [[action figure]]s and other merchandise of the character cover his genital area with part of his [[handgun holster]]. In the episode Vowel Play, it is implied that Assy has undersized genitals when he indicates that he can only give a school sex education teacher &quot;four and half inches&quot;. It is also unclear how he can pick up or handle anything as he has no hands. Items he is 'holding' simply float in front of him.  One commonly &quot;floating&quot; item seen around Assy is his revolver, which appears to mimic a Colt Python or Colt King Cobra. In addition to his police duties, Assy owns a used car, hot tub, and driveway repair dealership off &quot;Exit 19&quot; in Exeter which he does a commercial for wearing a cowboy hat and boots. He claims to have all the new {{not a typo|[[Scion xB]]s}} in stock, as well as having a true price guarantee and an unwillingness to give cars to people without credit (&quot;Bad credit? No credit? Sorry, that's your problem, asshole!&quot;) His slogan for the business is &quot;Don't make me run your ass over!&quot; Appearing only in season two, the &quot;commercial&quot; aired midway through each episode during the season and was part of a promotional arrangement between Adult Swim and Toyota, the parent company of Scion.&lt;ref&gt;{{Cite web |last=Lafayette |first=Jon |date=2008-04-03 |title=Scion Takes a Spin with 'Assy' |url=http://www.tvweek.com/news/2008/04/scion_takes_a_spin_with_assy.php |url-status=dead |archive-url=https://web.archive.org/web/20080407062026/http://www.tvweek.com/news/2008/04/scion_takes_a_spin_with_assy.php |archive-date=2008-04-07 |access-date=2012-02-22 |publisher=TVWeek}}&lt;/ref&gt;
1237 | caption = [[Sylvester Stallone]] as Rocky Balboa in &lt;br&gt; ''[[Rocky III]]'' (1982)
1238 | creator = [[Sylvester Stallone]]
1239 | portrayer = [[Sylvester Stallone]]
1240 '''Robert''' &quot;'''Rocky'''&quot; '''Balboa''' (born July 6, 1945{{citation needed|date=September 2024}}) (also known by his [[ring name]] the '''Italian Stallion''') is a fictional character and the titular [[protagonist]] of the [[Rocky (franchise)|''Rocky'' film series]]. The character was created by [[Sylvester Stallone]], who has also portrayed him in eight of the nine films in the franchise. He is depicted as a [[working class]] or poor [[Italian Americans|Italian-American]] from the slums of [[Philadelphia]] who started out as a [[club fighter]] and &quot;enforcer&quot; for a local [[Philadelphia Mafia|Philly Mafia]] [[loan shark]]. He is portrayed as overcoming the obstacles that had occurred in his life and in his career as a professional [[Boxer (boxing)|boxer]].
1241 The character is widely considered to be Stallone's most iconic role and is often considered the role that started his film career. He received critical acclaim for his performance in the first movie, earning [[Academy Award]] and [[Golden Globe Award]] nominations. When Stallone reprised his role once again in 2015 for ''[[Creed (film)|Creed]]'', his performance received universal acclaim and he received his first [[Golden Globe Award]] for [[Golden Globe Award for Best Supporting Actor - Motion Picture|Best Supporting Actor]], along with his third&lt;ref&gt;{{cite web |last1=King |first1=Susan |title=Golden Globes: Sylvester Stallone wins by a knockout |url=https://www.latimes.com/entertainment/movies/moviesnow/la-et-mn-golden-globes-stallone-20160110-story.html#:~:text=Sylvester%20Stallone%E2%80%99s%20Rocky%20Balboa%20made%20the%20comeback%20of,outing%20as%20the%20beloved%20Philadelphia%20pugilist%20in%20%E2%80%9CCreed.%E2%80%9D?adlt=strict&amp;toWww=1&amp;redig=9A4E443E50974C79AD1D7212FEBE2081 |website=[[Los Angeles Times]] |date=January 11, 2016 |access-date=1 August 2022}}&lt;/ref&gt; Oscar nomination for [[Academy Award for Best Supporting Actor|Best Supporting Actor]], the [[National Board of Review Award for Best Supporting Actor]] and several other accolades.
1242 The brain damage Rocky is diagnosed within ''Rocky V'' is not addressed in this film, but in interviews, Stallone has said that the storyline explanation would have been that Rocky's brain damage was within the normal range for boxers. When tested for [[brain damage]] in ''Rocky V'', Rocky was suffering the effects of a severe [[concussion]] as a result of the Drago fight, but he never sought a second or more informed opinion because he intended to retire anyway.&lt;ref&gt;{{cite web|url=http://www.aintitcool.com/node/30861 |title=Round One With Sylvester Stallone Q&amp;A!! |website=Aintitcool.com |access-date=December 1, 2015}}&lt;/ref&gt;
1243 Wanting to train in the old-school style, Donnie moves in with Rocky, staying in Paulie's former room. Donnie notices an old picture of Rocky and his son, Robert (an actual picture of [[Sylvester Stallone]] and a young [[Sage Stallone]]), Rocky reveals that Robert had moved to [[Vancouver]] with his girlfriend, because of the difficulties he faced trying to be independent in Philadelphia, but does check on his father every now and then. Pete, who initially wanted Rocky to be a part of his son, Leo's ([[Gabriel Rosado]]) team, challenges Donnie to fight his son, in which Rocky shows reluctance again, but then both agree.
```



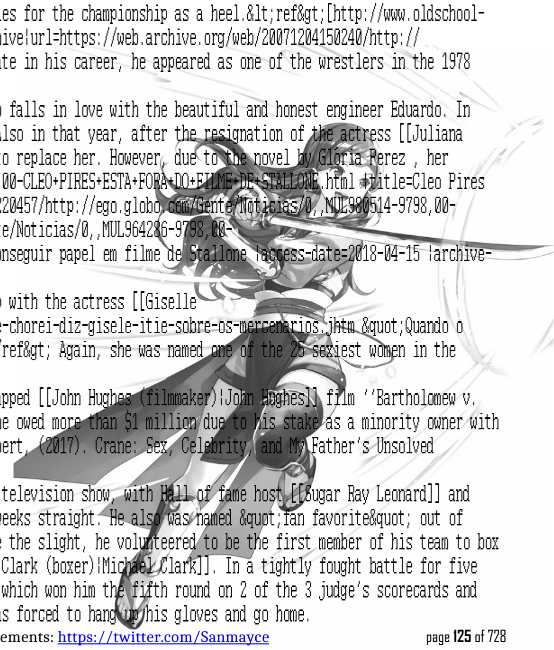
```
1244 Stallone also said there are &quot;ongoing discussions&quot; about a Rocky prequel television series, which he hopes will land on a streaming service and the series will likely follow a young Rocky Balboa as professional boxing hopeful. Stallone said Winkler is hesitant on making the series saying that &quot;There was some conflict there, yes. He felt in his mind that ''Rocky'' was primarily a feature film, and he did not see it as being translated for cable, so there was a big bone of contention.&quot;&lt;ref&gt;{{cite web|url=https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/ |title=Sylvester Stallone Feels Robbed of an Ownership Stake in 'Rocky': 'I Was Furious' |work= [[Variety (magazine)|Variety]] |date=July 23, 2019 |access-date=July 23, 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://popculture.com/movies/2019/07/23/sylvester-stallone-reveals-rocky-sequel-prequel-in-development/ |title=Sylvester Stallone Reveals 'Rocky' Sequel and Prequel Are in Development |work= popculture.com |access-date=July 23, 2019}}&lt;/ref&gt;
1245 That night, Rocky Balboa was born. People looked on him as the all-American tragedy, a man without much mentality and few social graces. But he has deep emotion and spirituality and good patriotism. And he has a good nature, although nature has not been particularly good to him. I have always seen him as a 20th Century gladiator in a pair of sneakers. Like so many of us, he is out of sync with the times. To all this, I injected doses of my own personal life, of my frustration at not getting anywhere.|30px|30px|Sylvester Stallone&lt;ref&gt;{{cite web|url=http://www.totalrocky.com/films/rocky/production.shtml |title=Rocky: Behind the Scenes Rocky (1976) &amp;#124; Making Of Rocky |website=Totalrocky.com |date=2014-06-20 |access-date=2015-12-01}}&lt;/ref&gt;}}
1246 Rocky Balboa fights as a [[Southpaw (boxing)|southpaw]] (left-handed). In the second film, against Apollo Creed, he comes out [[Orthodox (boxing)|orthodox]] and Mickey intends for him to switch back to southpaw late in the last round, but Balboa refuses saying &quot;no tricks, I ain't switching&quot;.  Mickey tells him that Apollo is ready for him (if he continues using his right) and so towards the end of the round, he does indeed lead with his left. The real reason for this is Sylvester Stallone tore his pectoral muscles in training,{{citation needed|date=December 2014}} but the idea was probably taken from the great southpaw boxer [[&quot;Marvelous&quot; Marvin Hagler]] who would sometimes come out orthodox to confuse opponents.{{citation needed|date=January 2021}}
1247 In 2011, Sylvester Stallone was inducted into the [[International Boxing Hall of Fame]] for his work on the Rocky Balboa character, having &quot;entertained and inspired boxing fans from around the world&quot;. Additionally, Stallone was awarded the Boxing Writers Association of America award for &quot;Lifetime Cinematic Achievement in Boxing.&quot;&lt;ref&gt;{{cite web|url=http://www.ibhof.com/pages/about/inductees/observer/stallone.html |title=Sylvester Stallone |publisher=International Boxing Hall of Fame |access-date=October 30, 2012}}&lt;/ref&gt; Inspired by people criticizing an actor being inducted in the Hall of Fame for playing a fictitious athlete, in 2014 the [[Fictitious Athlete Hall of Fame]] was launched with Rocky Balboa as its Inaugural Induction.&lt;ref&gt;{{cite web|url=http://www.notinhalloffame.com/fahof-current-inductees/3101-rocky-balboa|title=Rocky Balboa|website=Fictitious Athlete Hall of Fame|access-date=May 4, 2016}}&lt;/ref&gt;&lt;ref&gt;{{cite news|last1=Radley|first1=Scott|title=RADLEY: A Hall of Fame natural|url=http://www.thespec.com/sports-story/6880088-radley-a-hall-of-fame-natural/|website=[[The Hamilton Spectator]]|publisher=MetroLand Media|date=September 27, 2016}}&lt;/ref&gt;
1248 {{Sylvester Stallone}}
1249 [[Category:Sylvester Stallone]]</text>
1250     <redirect title="Sylvester Stallone" />
1251       <text bytes="31" sha1="kg7rj3t3splw2tq9hbup7dzvwfy63pv" xml:space="preserve">#REDIRECT[[Sylvester Stallone]]</text>
1252 * Xenon was the club used during the dance scene in ''[[Nighthawks (1981 film)|Nighthawks]]'' starring [[Sylvester Stallone]] and [[Billy Dee Williams]].
1253 ''''Torrente, the Dumb Arm of the Law''''' ({{lang-es|'''Torrente, el brazo tonto de la ley'''|links=no}})&lt;ref&gt;{{Cite book|first=Santiago|last=Juan-Navarro|chapter=''Torrente, el brazo tonto de la ley'' (''Torrente, the Dumb Arm of the Law'') (1998)|page=377-378|title=The Encyclopedia of Contemporary Spanish Films|editor-first=Salvador|editor-last=Jimenez Murguía|editor-first2=Alex|editor-last2=Pinar|publisher=[[Rowman &amp; Littlefield]]|isbn=9781442271333|year=2018}}&lt;/ref&gt;{{refn|group=n.|The Spanish original title is a parody of {{lang|es|Cobra, el brazo fuerte de la ley}} ({{translation|Cobra, the strong arm of the law}}), the title under which the 1986 [[Sylvester Stallone]]-starred film ''[[Cobra (1986 film)|Cobra]]'' was theatrically released in Spain.}} is a 1998 Spanish [[Black comedy|dark comedy]] film written and directed by [[Santiago Segura]], who stars as [[José Luis Torrente]], a racist, sexist, homophobic, xenophobic, and fascist former police agent.{{Sfn|Egea|2020|p=83}} Characterized by its deliberately cartoonish humor, it proved to be a massive box office hit, and Torrente became part of Spanish contemporary popular culture.
1254 **[[Sylvester Stallone]] - ''[[Rocky]]'' as [[Rocky Balboa]]
1255 **''[[Rocky]]'' - [[Sylvester Stallone]]
1256 |{{sortname|Muhammad|Ali}}&lt;br&gt;[[Sylvester Stallone]] || Presenters of the award for [[Academy Award for Best Supporting Actress|Best Supporting Actress]]
1257     <redirect title="Sylvester Stallone" />
1258       <text bytes="31" sha1="kg7rj3t3splw2tq9hbup7dzvwfy63pv" xml:space="preserve">#REDIRECT[[Sylvester Stallone]]</text>
1259 | scope=&quot;row&quot; | {{sortname|Dolly|Parton}}&lt;br&gt;[[Sylvester Stallone]]
1260 * [[Lock Up (1989 film)|''Lock Up'' (1989 film)]], an American prison action film featuring Sylvester Stallone
1261 A life member of the [[Actors Studio]],&lt;ref&gt;{{cite book|first=David |last=Garfield|title=A Player's Place: The Story of The Actors Studio|url=https://archive.org/details/playersplacestor00garf|url-access=registration|year=1980|publisher=MacMillan Publishing Co., Inc.|location=New York|isbn=0-02-542650-8|page=[https://archive.org/details/playersplacestor00garf/page/277 277]|chapter=Appendix: Life Members of The Actors Studio as of January 1980}}&lt;/ref&gt; Burton numbers among his many credits a co-starring role in ''[[Frank's Place]]'' and parts in films, such as ''[[Stir Crazy (film)|Stir Crazy]]'' and ''[[The Toy (1982 film)|The Toy]]''. He also appeared as Wells, one of the prisoners trapped in the besieged police station in [[John Carpenter]]'s 1976 [[Howard Hawks]]-inspired [[action film]], ''[[Assault on Precinct 13 (1976 film)|Assault on Precinct 13]]''. He later starred in the [[Rocky (film series)|''Rocky'' films]] as a trainer to [[Apollo Creed]] ([[Carl Weathers]]) and later [[Rocky Balboa]] ([[Sylvester Stallone]]). He appeared in an episode of ''[[Gibbsville (TV series)|Gibbsville]]'' in 1976. Burton also appeared in ''[[The Shining (film)|The Shining]]'', ''[[House Party 2]]'', and ''[[Hook (film)|Hook]]''.&lt;ref name=&quot;Tony Bio&quot;&gt;{{cite web|url=http://totalrocky.com/tony-burton-bio|access-date=February 27, 2016|work=Total Rocky|title=Tony Burton}}&lt;/ref&gt; He had [[guest appearance]]s in ''[[Kojak]]'', ''[[The Rockford Files]]'', ''[[CHiPs]]'', ''[[Twin Peaks]]'', and ''[[The A-Team]]''.&lt;ref name=&quot;Tony Bio&quot;/&gt; Burton also guest starred as Conrad King Baylor on ''[[In the Heat of the Night (TV series)|In the Heat of the Night]]'' in the episode &quot;King's Ransom&quot;, which aired in January 1990.
1262 | [[List of sports films|Sports]] drama film, directed by [[John G. Avildsen]], and both written by and starring [[Sylvester Stallone]].&lt;ref name=&quot;Rocky&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/20655/Rocky/|title=Rocky|work=TCM database|publisher=[[Turner Classic Movies]]|access-date=February 28, 2016}}&lt;/ref&gt;
1263 * Written and directed by, and starring [[Sylvester Stallone]].
1264 * Sports-drama film, written and directed by, and starring Sylvester Stallone.&lt;ref name=&quot;Rocky III&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/20807/Rocky-III/|title=Rocky III|work=TCM database|publisher=[[Turner Classic Movies]]|access-date=February 28, 2016}}&lt;/ref&gt;
1265 * Sports film, written and directed by, and starring Sylvester Stallone.&lt;ref name=&quot;Rocky IV&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/20886/Rocky-IV/|title=Rocky IV|work=TCM database|publisher=[[Turner Classic Movies]]|access-date=February 28, 2016}}&lt;/ref&gt;
1266 * Sports drama film, written by Sylvester Stallone.&lt;ref name=&quot;Rocky V&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/21906/Rocky-V/|title=Rocky V|work=TCM database|publisher=[[Turner Classic Movies]]|access-date=February 28, 2016}}&lt;/ref&gt;
1267 * Sports drama film, written and directed by, and starring [[Sylvester Stallone]].&lt;ref name=&quot;Rocky Balboa&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/636512/Rocky-Balboa/|title=Rocky Balboa|work=TCM database|publisher=[[Turner Classic Movies]]|access-date=February 28, 2016}}&lt;/ref&gt;
1268 |description=1929 [[Rapp and Rapp]] theater, unusually spacious for the area, closed in 1977. [[Sylvester Stallone]] once worked as an usher.
1269 |caption = North American NES cover artwork featuring [[Sylvester Stallone]] as [[John Rambo]]
1270 Later in the 1980s, Strange went to [[Ibiza]], Spain, and became an integral part of the budding [[Trance music|trance]] club movement and hosting parties for celebrities such as [[Sylvester Stallone]]. In the early 1990s, he was the host at the &quot;Double Bass&quot; club in Ibiza.
1271 | creator = [[Sylvester Stallone]]
1272 Former champion [[Apollo Creed]] ([[Carl Weathers]]), now 43 years old, comes out of retirement to challenge Drago to an exhibition match, assisted by Creed's former rival [[Rocky Balboa]] ([[Sylvester Stallone]]). Creed arrives to the ring wearing his signature [[Flag of the United States|Stars &amp; Stripes]] boxing garb to &quot;[[Living in America (James Brown song)|Living in America]]&quot;, sung by [[James Brown]], dancing upon a huge stage that is lowered into the ring. Before the match begins, Drago mutters &quot;You will lose.&quot;
1273 Melchior's 1956 short story &quot;The Racer&quot; was adapted as [[Paul Bartel]]'s [[cult film]] favorite, ''[[Death Race 2000]]'' (1975), starring [[David Carradine]] and [[Sylvester Stallone]] and produced by [[Roger Corman]]. It was later remade as ''[[Death Race (2008 film)|Death Race]]'' (2008), starring [[Jason Statham]] and [[Joan Allen]], directed by [[Paul W. S. Anderson]] and produced by [[Tom Cruise]].
1274 He co-produced ''[[Rhinestone (film)|Rhinestone]]'', starring [[Dolly Parton]] and [[Sylvester Stallone]], for [[20th Century Fox]] as well as other films, prior to 1982.
1275 :It's definitely possible. Depends what your goal for bulk is, but with enough protein and exercise, one can add major muscle mass without resorting to drugs.  You may not end up looking like [[Lou Ferrigno|the hulk]], but [[Sylvester Stallone|rambo]] is in reach, given the right genes. [[User:Night Gyr|Night Gyr]] 17:35, 7 April 2006 (UTC)
1276 |[[Sylvester Stallone]] || ''[[Creed (film)|Creed]]'' || [[Rocky Balboa]]
1277 A later attempt to adapt ''The Executioner'' to the screen by [[William Friedkin]] was to star [[Sylvester Stallone]] and [[Cynthia Rothrock]],&lt;ref&gt;[http://www.hkmania.com/from.php?fromurl=http://www.hkmania.com/Dossiers/cynthiarothrockeng.html Hk Mania - Le site des fans de cinéma asiatique]&lt;/ref&gt; but the production was scrapped.&lt;ref&gt;[http://www.speakeasy.org/~swmpthng/StonyMan.html The Stony Man Farm]&lt;/ref&gt;
1278 | creator     = [[Sylvester Stallone]]
1279 '''Apollo Creed''' is a fictional [[Character (arts)|character]] from the [[Rocky (franchise)|''Rocky'' franchise]], played by [[Carl Weathers]]. He appeared in the first four ''Rocky'' films. Creed is the main [[antagonist]] in ''[[Rocky]]'' and ''[[Rocky II]]'', and one of the [[protagonists]] in ''[[Rocky III]]'' and ''[[Rocky IV]]''. [[Sylvester Stallone]], the writer and star of ''Rocky'', said &quot;[[Jack Johnson (boxer)|Jack] Johnson]] served as the inspiration for the character of Apollo Creed.&quot;&lt;ref&gt;{{Cite news |url=https://www.reuters.com/article/us-usa-trump-pardon/trump-pardons-late-black-boxing-champion-jack-johnson-idUSKCN1IP31F |title=Trump pardons late black boxing champion Jack Johnson |publisher=Reuters |access-date=24 May 2018}}&lt;/ref&gt;
1280 :Note: Not only has this mini-series seen an English translation as ''Bullet to the Head'' by [[Dynamite Entertainment]] with each of the three French originals split into two US [[comic book]]s, but it had also been the basis for the 2012 [[Silvester Stallone]] film of similar title as already above-mentioned. The February 2013 reprint of the French intégrale edition ({{ISBN|978-2-203-06073-9}}) sported a new cover featuring the countenance of Stallone on its cover.
1281 As well as performing monologue comedy segments on ''The Comedy Company'', Mole interviewed various celebrities, including [[Julian Lennon]], [[Sylvester Stallone]], the band [[INXS]] and [[Kylie
```



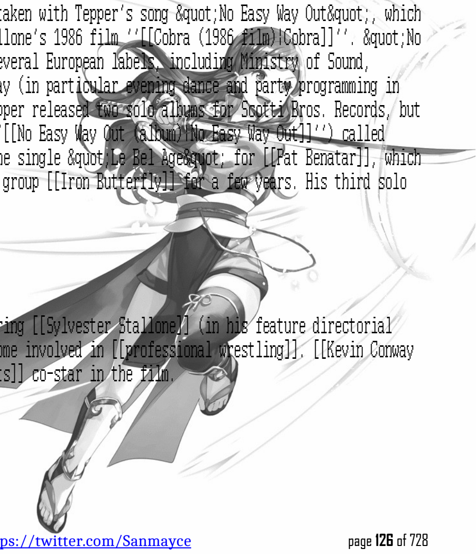
```
Minogue]].
1282 All of Vinny's training has been for an upcoming fight. At one point during his training he runs up the [[Rocky Steps]], imitating [[Sylvester Stallone]]'s famous climb. The music video culminates with a fight between Vinny and another boy in front of a packed crowd. Some of the footage seen is of an actual fight that Vinny fought in at [[The Blue Horizon]]. When the &quot;junior match&quot; gets underway, Vinny comes out punching.  Between each of the punches, the video flashes back to images from Vinny's father's death. These images include: his father's fighter plane getting shot down over Iraq, his mother picking up the heart-breaking phone call, and the military funeral ceremony. The entire segment ends with Vinny winning the fight and celebrating with the crowd. The video parallels the song by showing the ways in which people find ways to forget about and move on from bad things.  In this case Vinny leaves behind his toy planes which remind him of his father and boxes to forget about the painful event of his death.
1283 *[[Norman Jewison]]'s ''[[F.I.S.T. (film)¦F.I.S.T.]]'', loosely based on Teamsters President [[Jimmy Hoffa]], starring [[Sylvester Stallone]], [[Rod Steiger]] and [[Peter Boyle]] and [[Melinda Dillon]] is released.
1284 Hood sold his first spec screenplay to [[MTV Films]] in 2000, and went on to sign a deal with [[Dimension Films]],&lt;ref&gt;Variety (2001) [https://www.variety.com/article/VR1117851315.html?categoryid=13&amp;cs=1&amp;query=Hood+Dimension Variety] Retrieved October 21, 2010&lt;/ref&gt; which included rewrites on ''[[Halloween: Resurrection]]'' and ''[[Cursed (2005 film)¦Cursed]]''. He went on to work on ''[[Conan the Barbarian (2011 film)¦Conan the Barbarian]]'' for producer [[Avi Lerner]],&lt;ref&gt;Variety (2007) [https://www.variety.com/article/VR1117966004.html?categoryid=13&amp;cs=1 Variety] Retrieved June 26, 2009&lt;/ref&gt; and ''[[Hercules: The Legend Begins]]''. Most recently, he penned an screenplay adaptation of the novel Rolling in The Deep, which will be produced by [[J. Todd Harris]] and directed by [[Mary Lambert (director)]].&lt;ref&gt;Variety (2018) [https://variety.com/2018/film/news/mary-lambert-mermaid-thriller-rolling-in-the-deep-1203029736/ Variety] Retrieved November 15, 2018&lt;/ref&gt; In 2011, Hood was hired to write the script for the fifth ''Rambo'' film, titled ''Rambo: Last Stand'',&lt;ref&gt;{{cite web¦url=https://www.empireonline.com/movies/news/new-screenplay-rambo-5/¦title=New Screenplay For Rambo 5¦first=Owen¦last=Williams¦work=Empire¦date=August 19, 2011¦access-date=October 2, 2018¦url-status=live¦archive-date=April 2, 2019¦archive-url=https://web.archive.org/web/20190402011517/https://www.empireonline.com/movies/news/new-screenplay-rambo-5/}}&lt;/ref&gt; however, Hood's script was put on hold in early 2012.&lt;ref&gt;{{cite web¦url=https://www.joblo.com/horror-movies/news/sean-hood-finally-gives-an-update-on-the-state-of-rambo-5¦title=Sean Hood Finally Gives an Update on the State of Rambo 5¦first=Marcey¦last=Papandrea¦work=Joblo¦date=February 10, 2012¦access-date=October 2, 2018}}&lt;/ref&gt; In 2014, Hood's ''Rambo: Last Stand'' script was abandoned in favor of a new script by [[Sylvester Stallone]].&lt;ref&gt;{{cite web¦url=https://theplaylist.net/no-country-for-old-men-esque-rambo-v-reportedly-on-the-way-20140623/¦title='No Country for Old Men'-Esque 'Rambo V' Reportedly On The Way¦first=Kevin¦last=Jagernauth¦work=The Playlist ¦date=June 23, 2014¦access-date=August 27, 2021¦url-status=live¦archive-date=August 28, 2021¦archive-url=https://web.archive.org/web/20210828030403/https://theplaylist.net/no-country-for-old-men-esque-rambo-v-reportedly-on-the-way-20140623/}}&lt;/ref&gt;
1285 The 1980s saw the theme reworked in movies such as Missing in Action (1984) and especially in the three Rambo films that starred Sylvester Stallone. The second in the latter series, Rambo: First Blood, Part II (1985), showed the hero returned to Vietnam after the conflict to free American soldiers still held as prisoners of war. It was an audience favorite despite a generally acknowledged a lack of realism. Later in the 1980s, movies such as director Oliver Stone's Platoon (1986) brought a more realistic portrayal of the experience of American soldiers in Vietnam.
1286 '''''Demolition Man''''' is an [[Extended play¦EP]] released by [[A&amp;M Records¦A&amp;M]] on September 21, 1993 ([[United States]]) and November 1, 1993 ([[United Kingdom¦UK]]) in support of the [[Sylvester Stallone]]/[[Wesley Snipes]] film ''[[Demolition Man (film)¦Demolition Man]]''. It features Sting's re-recording of [[The Police]] track &quot;[[Demolition Man (song)¦Demolition Man]]&quot;, as well as several live recordings. The live recordings were recorded at [[Villa Manin]], [[Codroipo]], [[Italy]] on 25 July 1993. The US&lt;ref&gt;[http://www.discogs.com/Sting-Demolition-Man/release/757051 Sting - Demolition Man (CD, MiniAlbum, US) at Discogs]&lt;/ref&gt; and International/European&lt;ref&gt;[http://www.discogs.com/Sting-Demolition-Man/release/1006606 Sting - Demolition Man (CD, MiniAlbum, Europe) at Discogs]&lt;/ref&gt; [[Extended play¦EP]] release features the title track and five live recordings and sales for this release were listed on the [[Billboard 200¦US Albums Chart]], as the UK release&lt;ref&gt;[http://www.discogs.com/Sting-Demolition-Man/release/862216 Sting - Demolition Man (CD, Maxi-Single, UK) at Discogs]&lt;/ref&gt; was a standard [[Single (music)¦single]] which featured an exclusive single version of the title track as well as the album version and two live recordings. The UK release charted therefore on the [[UK Singles Chart]], where it peaked at number 21.&lt;ref&gt;[https://www.officialcharts.com/search/singles/Demolition%20Man Official Charts Company - Sting - Demolition Man]&lt;/ref&gt;
1287 ¦ caption = North American SNES cover artwork featuring [[Sylvester Stallone]] as Sergeant John Spartan and [[Wesley Snipes]] as Simon Phoenix
1288 The 3DO game was developed in tandem with the film&lt;ref&gt;{{cite magazine¦title=Demolition Man¦magazine=[[GamePro]]¦issue=76¦publisher=[[International Data Group¦IDG]]¦date=January 1995¦page=192}}&lt;/ref&gt; and entails several gameplay styles: of the game's 16 levels, nine are [[light gun shooter]] levels, four are [[Fighting game¦one-on-one fighting]], two are [[first-person shooter]]; and one is a mission-based [[racing game]]. Due to the advanced 3DO sound engine, the [[Demolition Man (soundtrack)¦soundtrack]] is ripped directly from the movie, and [[full motion video]] is heavily used, incorporating both footage from the film and exclusive footage of [[Sylvester Stallone]] and [[Wesley Snipes]] made specifically for the game. [[Jesse Ventura]] reprises his role as a minion to Simon Phoenix as the only actor in the movie to play a cryo-con henchman in the underground level and cryo-prison level.{{citation needed¦date=July 2016}} A version of the game was in development by Virgin Interactive and planned to be published by [[Atari Corporation]] for the [[Atari Jaguar CD]],&lt;ref&gt;{{cite magazine¦url=https://archive.org/stream/nextgen-issue-006/Next_Generation_Issue_006_June_1995#page/n19¦title=Jaguar Plugs into the CD Revolution¦magazine=[[Next Generation (magazine)¦Next Generation]]¦issue=6¦publisher=[[Imagine Media]]¦date=June 1995¦page=19}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine¦url=https://archive.org/stream/GamePro_Issue_073_August_1995#page/n38/mode/1up¦title=Electronic Entertainment Expo Special Report - Jaguar on the Prowl¦magazine=[[GamePro]]¦issue=83¦publisher=[[International Data Group¦IDG]]¦date=August 1995¦page=37}}&lt;/ref&gt; but it was cancelled after several delays.&lt;ref&gt;{{cite web¦url=http://www.atarimania.com/game-atari-jaguar-creature-shock_31083.html¦title=Atari Jaguar - Creature Shock¦website=atarimania.com¦accessdate=2018-09-11¦archive-url=https://web.archive.org/web/20180911225305/http://www.atarimania.com/game-atari-jaguar-creature-shock_31083.html¦archive-date=2018-09-11¦url-status=live}}&lt;/ref&gt; A [[PlayStation (console)¦PlayStation]] version was also in development but never released.&lt;ref&gt;{{cite magazine¦url=https://retrocdn.net/index.php?title=File%3AHyper_AU_011.pdf&amp;page=8¦title=News - The Sony PlayStation - Sega and Sony go Head to Head¦magazine=[[Hyper (magazine)¦Hyper]]¦publisher=[[nextmedia¦Next Media Pty Ltd]]¦issue=11¦date=October 1994¦page=8}}&lt;/ref&gt;
1289 In the 2022 [[Paramount+]] television series ''[[Tulsa King]]'', main character Dwight &quot;The General&quot; Manfredi, portrayed by [[Sylvester Stallone]], resides in the Mayo Hotel.&lt;ref&gt;{{cite web¦title = 'Tulsa King': The Mayo Hotel Is a Real Place in Oklahoma¦url =https://www.msn.com/en-us/travel/tripideas/tulsa-king-the-mayo-hotel-is-a-real-place-in-oklahoma/ar-AA146Yp8¦publisher=Showbiz Cheatsheet¦ access-date =2022-12-25 }}&lt;/ref&gt;
1290 {{blockquote¦&quot;The desirability of hiring directors over 60 is fairly diminished in this marketplace. At the same time, I hadn't had a good-sized hit in quite a while. And, frankly, I went through a couple of experiences that left me pretty disgusted with it all, and I was thinking the time had passed. I was just sitting at home reading magazines and looking out the window - a couple of projects I had had just fallen apart - when I got a call from Sly, who had sent me a script.&quot;&lt;ref&gt;{{cite news ¦url=http://www.laweekly.com/film/the-great-action-director-walter-hill-attempts-a-comeback-and-brings-along-sylvester-stallone-2612773 ¦first=Nick ¦last=Pinkerton ¦title=The Great Action Director Walter Hill Attempts a Comeback (and Brings Along Sylvester Stallone) ¦newspaper=[[LA Weekly]] ¦date=January 31, 2013 ¦access-date=26 April 2015}}&lt;/ref&gt;}}
1291 It was [[Sylvester Stallone]] who offered Hill the job of directing ''[[Bullet to the Head]]'' (2012). The film was not a success at the box office.
1292 The soundtrack was hugely successful on the strength of two top-five singles, [[Survivor (band)¦Survivor]]'s &quot;[[Burning Heart (song)¦Burning Heart]]&quot; (personally commissioned for the film by [[Sylvester Stallone]]) reached No. 2 on the [[Billboard Hot 100¦''Billboard'' Hot 100]])&lt;ref name=&quot;Songfacts&quot;&gt;{{cite web
1293 * [[Frank Stallone Sr.]], father of [[Sylvester Stallone]], born in Gioia del Colle before emigrating to the United States
1294 *[[Sylvester Stallone]]
1295 The building remained un[[listed building¦listed]]. The building was featured in the [[Channel 4]] series ''[[Demolition (television)¦Demolition]]'' in [[2005 in television¦2005]]. At the same time general interest in the car park increased, partly as a reflection of the recognition of ''Get Carter'' as a classic of [[British cinema]], with [[Sylvester Stallone]] lending his [[Get Carter (2000 film)¦weight]] to the calls for it to be preserved as a cinematic landmark.
1296 [[Category:Images of Sylvester Stallone]]</text>
1297 *[[Sylvester Stallone]]
1298 '''''Lock Up''''' is a 1989 American [[prison film¦prison]] [[action film]]&lt;ref&gt;{{cite web¦url=https://goodmenproject.com/arts/prison-action-film-lock-up-4k-ultra-hd-jsnk/¦title=The Prison Action Film 'Lock Up' is Coming to 4K Ultra HD¦website=[[The Good Men Project]]¦date=September 10, 2019¦access-date=July 27, 2021}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=https://www.allmovie.com/movie/lock-up-v29821¦title=Lock Up (1989) - John Flynn¦website=[[AllMovie]]¦access-date=July 27, 2021}}&lt;/ref&gt; directed by [[John Flynn (director)¦John Flynn]], and starring [[Sylvester Stallone]], [[Donald Sutherland]], [[John Amos]] and [[Tom Sizemore]]. It was released in the United States on August 4, 1989.
1299 *[[Sylvester Stallone]] as Frank Leone
1300 *''[[Escape Plan (film series)]]'', another Sylvester Stallone film series set in a prison.
1301 '''''Rhinestone''''' is the soundtrack album from the [[Rhinestone (film)¦1984 film of the same name]] starring [[Dolly Parton]] and [[Sylvester Stallone]]. It was released on June 18, 1984, by [[RCA Victor]]. The album was produced  by [[Mike Post]] and Parton. It peaked at number 22 on the ''[[Billboard (magazine)¦Billboard]]'' [[Top Country Albums]] chart and number 135 on the [[Billboard 200¦''Billboard'' 200]]. The Dolly Parton-composed soundtrack produced two top ten singles on the ''Billboard'' [[Hot Country Songs]] chart: &quot;Tennessee Homesick Blues&quot; and &quot;God Won't Get You&quot;, which peaked at numbers one and ten, respectively.
1302 ¦ extra7 = [[Sylvester Stallone]] with Dolly Parton
1303 ¦ extra8 = Dolly Parton and Sylvester Stallone
1304 ¦ extra10 = Sylvester Stallone
1305 ¦ extra11 = Dolly Parton and Sylvester Stallone
1306 ¦ extra14 = Dolly Parton and Sylvester Stallone
1307 *[[Sylvester Stallone]] - lead vocals
1308 ¦ starring      = [[Sylvester Stallone]]&lt;br /&gt;[[Demi Moore]]&lt;br /&gt;[[Steven Spielberg]]&lt;br /&gt;[[Michael J. Fox]]&lt;br /&gt;[[Brian Grazer]]
1309 The film is also notable for its use of a number of celebrity actors, musicians, writers, producers, and directors in the Universal family, including [[Andrew Bergman]], [[James Cameron]] (shown improving the Universal Studios landscaping), [[Shaun Cassidy]], [[Robin Cook (novelist)¦Robin Cook]], [[Shelley Fabares]], [[Michael J. Fox]], [[Brian Grazer]], [[Heavy D]], [[Jeffrey Katzenberg]], [[Barry Kemp (TV producer)¦Barry Kemp]], [[Angela Lansbury]] (shown painting the ''[[Psycho (1960 film)¦Psycho]]'' House),  Mike Lobell, [[Traci Lords]], [[Kevin Misher]], [[Demi Moore]], [[Darrin Pfeiffer]], [[John Singleton]], [[Steven Spielberg]] (shown as a Universal Studios Guide), [[Sylvester Stallone]] (in his [[Rocky Balboa (character)¦Rocky Balboa]] character, and subtitled for comedic effect), [[Marty Stuart]], [[David Zucker (director)¦David Zucker]], and [[Jerry Zucker]].
1310 It was nominated for five [[Razzie Awards]] including Worst Picture (which lost to ''[[Cocktail (1988 film)¦Cocktail]]''), Worst Actor (Bobcat Goldthwait; who lost to [[Sylvester Stallone]] for
```



```
''[[Rambo III]]''), Worst Director (Michael Dinner; who lost to both [[Blake Edwards]] for ''[[Sunset (1988 film)|Sunset]]'' and [[Stewart Raffill]] for ''[[Mac and Me]]'' in a tie), Worst Screenplay (also lost to ''Cocktail'') and Worst New Star (Don the talking horse; who lost to [[Ronald McDonald]] for his cameo in ''Mac and Me'').{{cn|date=July 2024}}
*''[[Oscar (1991 movie)|Oscar]]'' (film) [https://www.imdb.com/title/tt0102603/] starring [[Peter Riegert]], [[Chazz Palminteri]], [[Joey Travolta]] and [[Sylvester Stallone]]
* [[Sylvester Stallone]] (born 1946), actor and film producer; mother is half French; maternal grandmother is from [[Brest, France|Brest]]
|[[Sylvester Stallone]] and [[Jamiroquai]] episode
&quot;Rhinestone Cowboy&quot; served as the basis for the film ''[[Rhinestone (film)|Rhinestone]]'' (1984), starring [[Sylvester Stallone]] and [[Dolly Parton]]
&quot;When you talk about actors, Dustin Hoffman and Laurence Olivier are actors&quot;, said Norris. &quot;They can do anything. Then you have your personalities, Burt Reynolds, Sylvester Stallone, Charles Bronson, Clint Eastwood, and me. When they deviate too much from what audiences expect, they don't do very well, do they?&quot;&lt;ref&gt;{{cite news |last=Maslin |first=Janet |date=27 Apr 1985 |title=Norris no Olivier but he keeps busy |work=The Globe and Mail |page=E.3}}&lt;/ref&gt;
In 2015, Warburtons announced the use of [[Sylvester Stallone]] in its marketing campaign, with Stallone re-enacting parts he had played previously.&lt;ref&gt;{{Cite news |last=Emma Flanagan |date=13 April 2015 |title=Watch Sylvester Stallone star in new Warburtons advert 'The Deliverers' filmed in Bolton |newspaper=[[Manchester Evening News]]|url=http://www.manchestereveningnews.co.uk/whats-on/film-news/watch-sylvester-stallone-star-new-9034511}}&lt;/ref&gt; Another advert involves [[List of Muppets|The Muppets]] singing about the new brand of giant crumpets, in a style almost identical to the opening theme of ''[[The Muppet Show]]'', whilst the advert with [[Peter Kay]] plays on the Bolton connection.{{cn|date=March 2024}}
In 2016, it was announced that Sarpei would be a host on the [[Netflix]] [[reality show]] ''[[Ultimate Beastmaster]]''.&lt;ref&gt;{{cite web|url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|title=Netflix Orders Global Competition Series 'Ultimate Beastmaster' from Sylvester Stallone, Dave Broome|last=Prudon|first=Laura|work=Variety|date=9 May 2016|access-date=5 October 2016}}&lt;/ref&gt;
* [[Jackie Stallone]] - astrologer and mother of Sylvester Stallone, maternal Jewish descent
    <title>Sylvester Stalone</title>
    <redirect title="Sylvester Stallone" />
      <text bytes="32" sha1="bnfj12dj30potn771e5t1yex066t6ak" xml:space="preserve">#redirect [[Sylvester Stallone]]</text>
*The Fontainebleau is featured in the 1992 film ''[[The Bodyguard (1992 film)|The Bodyguard]]'' starring [[Whitney Houston]]. Also in 1992, the hotel appears in final scene of ''[[The Specialist]]'', an action film starring [[Sylvester Stallone]] and [[Sharon Stone]].
He was married to fellow animator Jean Washam,&lt;ref&gt;{{Cite news |title=Ben Washam (Or Mike Fontanelli?) |url=https://www.cartoonbrew.com/old-brew/ben-washam-or-mike-fontanelli-680.html |last=Amidi |first=Amid |access-date=2022-05-20}}&lt;/ref&gt; who he met at Warner Bros. in 1951 (Jean also had various stints working for [[Walt Disney Animation Studios|Walt Disney Productions]], [[Hanna-Barbera|Hanna-Barbera Productions]], and [[Shamus Culhane]]'s studio, among others) and married three years later.&lt;ref name=&quot;YNLNews&quot;&gt;{{cite news |last1=Rausch |first1=Emma |title=Former Wabash-native Recalls her Career as a Golden Age Animator |url=https://www.yournewslocal.com/former-wabash-native-recalls-her-career-as-a-golden-age-animator/ |access-date=May 23, 2022 |work=Your News Local |date=January 2, 2018 |archive-url=https://web.archive.org/web/20230524015932/https://www.yournewslocal.com/former-wabash-native-recalls-her-career-as-a-golden-age-animator/ |archive-date=May 24, 2023}}&lt;/ref&gt; They had a son named Trevor, who shoed horses for a living, and he notably shoed the horses of actors [[Sylvester Stallone]] and [[William Shatner]].&lt;ref name=Washam&gt;{{Cite news |title=Heir to a cartoon legacy hides free art in Snoqualmie Valley |url=https://www.king5.com/article/news/community/cartoon-free-art-snoqualmie-valley-bugs-bunny-grinch/281-2aa2cd8f-dbc1-4a43-b2ab-c818571d1686 |last=Erickson |first=Anne |date=2021-03-08 |access-date=2022-05-20 |work=[[KING-TV]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |url=https://livingsnoqualmie.com/snoqualmie-valley-artist-ben-washam-spreading-joy-through-art-during-the-pandemic/ |title=Snoqualmie Valley Artist Ben Washam: Spreading Joy Through Art During the Pandemic |date=2021-02-10 |access-date=2022-05-20 |website=Living Snoqualime |last=Grant |first=Melissa}}&lt;/ref&gt;
Nadiadwala released ''[[Heyy Babyy]]'' in 2007, wherein he launched director [[Sajid Khan (director)]].&lt;ref&gt;{{cite web |url=http://www.boxofficeindia.com/showProd.php?itemCat=214&amp;catName=MjAwNw== |title=Box Office 2007 |website=Boxofficeindia.com |access-date=27 December 2011 |url-status=dead |archive-url=https://web.archive.org/web/20131014132441/http://boxofficeindia.com/showProd.php?itemCat=214&amp;catName=MjAwNw%3D%3D |archive-date=14 October 2013 |df=dmy-all }}&lt;/ref&gt; His 2009 production ''[[Kambakkht Ishq]]'', for the very first time in Bollywood included the casting of Hollywood stars such as [[Sylvester Stallone]], [[Denise Richards]] and [[Brandon Routh]]. The film starred [[Akshay Kumar]] and [[Kareena Kapoor]] in the lead. Nadiadwala described the moment when he signed Sylvester Stallone for the film as historic since he was so star-struck by Stallone, as he was wondering whether he should get an autograph first or get the contract signed.&lt;ref&gt;
|title=Nadiadwala signs Sylvester Stallone for his next film |newspaper=The Economic Times
[[File:Rocky (1976) - Rocky Steps.ogv|thumb|Video of [[Sylvester Stallone]] as [[Rocky Balboa]] running up the front steps of the [[Philadelphia Museum of Art]] in [[Philadelphia]] in ''[[Rocky]]'' in 1976; the iconic scene gave birth to the steps' notoriety and popularity.]]
[[File:Sylvester Stallone, Tessa Thompson, and Michael B. Jordan promoting Creed at the Philadelphia Art Museum.JPG|thumb|[[Sylvester Stallone]], [[Tessa Thompson]], and [[Michael B. Jordan]] promoting ''[[Creed (film)|Creed]]'' atop the Rocky Steps in November 2015]]
The '''Rocky Steps''' are 72 stone steps leading up to the East entrance of the [[Philadelphia Museum of Art]] in [[Philadelphia]] that gained global fame after being featured in a notable scene from the 1976 film ''[[Rocky (film)|Rocky]]''. In the scene, [[Rocky Balboa]], an unpolished but ambitious boxer from [[South Philadelphia]] played by [[Sylvester Stallone]], begins intense physical training after deciding to fight [[Apollo Creed]], the [[World Heavyweight Champion]]. The scene is widely considered one of the most iconic in the history of modern films.&lt;ref&gt;[https://the-rearview-mirror.com/2022/09/14/cinemas-greatest-scenes-when-rocky-runs-up-the-steps-of-philadelphia-museum-of-art/ &quot;Cinemas Greatest Scenes: When Rocky runs up the steps of the Philadelphia Museum of Art&quot;], Rearview Mirror&lt;/ref&gt;&lt;ref&gt;[https://entertainment.ie/movies/movie-news/heres-the-story-behind-the-iconic-steps-steadicam-scene-from-rocky-234554/ &quot;Here's the story behind the iconic 'Steps' Steadicam scene from Rocky&quot;], Entertainment.ie]&lt;/ref&gt;
''Rocky'' creator [[Sylvester Stallone]] has recounted that the genesis of the iconic scene occurred when the 1976 film crew for the movie, constrained by a tight budget, identified the steps one night while searching for filming locations around the city. Stallone first thought Rocky should carry his dog Butkus up the steps, but the big [[bull mastiff]] proved too heavy for the scene to work. Still, the view from the top of the stairs inspired him to reshoot the scene without the dog. In the 2006 film ''[[Rocky Balboa (film)|Rocky Balboa]]'', Rocky lifts his dog Punchy when he reaches the top of the steps. The [[closing credits]] of ''Rocky Balboa'' show a montage of dozens of people running up the steps.
Prior to the 1982 release of ''[[Rocky III]]'', [[Sylvester Stallone]] commissioned [[A. Thomas Schomberg]] to create a bronze statue of Rocky.&lt;ref name=&quot;pophistorydig&quot;&gt;{{cite web | url=http://www.pophistorydig.com/?tag=rocky-statue-philadelphia-art-museum | title=The Rocky Statue | work=The Pop History Dig | date=2009-07-20 | author=Doyle, Jack}}&lt;/ref&gt; Three 2-ton, {{convert|10|foot|adj=on}}-tall copies were to be cast. One was installed atop the steps for the filming of ''Rocky III'', and was ultimately relocated to the bottom of the steps. The second ''Rocky'' was in the [[San Diego Hall of Champions|San Diego Hall of Champions Sports Museum]] in [[San Diego, California]], until it closed in 2017. The statue was then put up for auction, and purchased by an anonymous buyer who was later revealed to be Sylvester Stallone himself.
Canada eventually became home to Jonathan in the early 1960s as he settled in the [[Vancouver, British Columbia|Vancouver]] suburb of [[Langley, British Columbia (city)|Langley]]. Making Vancouver his home base, he competed frequently for [[NWA All Star Wrestling]], winning five Pacific Coast Heavyweight titles between 1970 and 1977, the NWA World Tag Team title (with [[Dominic Denucci]]) in 1966, and a record 18 Canadian Tag Team titles between 1964 and 1978, as well as challenging for the NWA World Heavyweight Championship against such titleholders as Kiniski, [[Dory Funk, Jr.]] and [[Jack Brisco]]; he also engaged in feuds with Kiniski and [[Dutch Savage]] in All Star, as well as teaming with them. On May 31, 1972, in what was billed as the &quot;match of the century,&quot; Jonathan defeated Le Géant Jean Ferré ([[André the Giant]]) by disqualification.&lt;ref name=slam/&gt; On September 7, 1972, in a match which was billed as the &quot;Battle of the Giants&quot; Jonathan had a rematch against André, this time losing by disqualification. In 1973 he wrestled in the WWWF and fought Pedro Morales for the championship as a heel.&lt;ref&gt;[http://www.oldschool-wrestling.com/geeklog/article.php/Andre-The-Giant-1 André the Giant vs. Don Leo Jonathan - May 31 and September 7, 1972] {{Webarchive|url=https://web.archive.org/web/20071204150240/http://www.oldschool-wrestling.com/geeklog/article.php/Andre-The-Giant-1 |date=December 4, 2007 }} at Old School Wrestling&lt;/ref&gt; Late in his career, he appeared as one of the wrestlers in the 1978 [[Sylvester Stallone]] movie ''[[Paradise Alley]]''.
In 2008, she co-starred in the [[remake]] of ''Ciranda de Pedra'', in which she lived the young teacher Margarida, a shy girl, who falls in love with the beautiful and honest engineer Eduardo. In 2009, she joined the cast of the soap opera ''[[Caminho das Índias]]'', in which she played the Indian Surya, her first villain. Also in that year, after the resignation of the actress [[Juliana Paes]] in acting in the long one ''[[The Expendables (2010 film)|The Expendables]]'', of [[Sylvester Stallone]], Cleo was chosen to replace her. However, due to the novel by Glória Perez , her participation in the film could not be fulfilled.&lt;ref&gt;{{Cite web |url=http://ego.globo.com/Gente/Noticias/0,,MUL980514-9798,00-CLEO+PIRES+ESTA+FORA+DO+FILME+DE+STALLONE.html |title=Cleo Pires está fora do filme de Stallone |access-date=2018-04-15 |archive-date=2016-03-03 |archive-url=https://web.archive.org/web/20160303220457/http://ego.globo.com/Gente/Noticias/0,,MUL980514-9798,00-CLEO+PIRES+ESTA+FORA+DO+FILME+DE+STALLONE.html |url-status=dead }}&lt;/ref&gt;&lt;ref&gt;{{Cite web |url=http://ego.globo.com/Gente/Noticias/0,,MUL964286-9798,00-QUASE+LA+CLEO+PIRES+SAI+NA+FRENTE+PARA+CONSEGUIR+PAPEL+EM+FILME+DE+STALLONE.html |title=Quase lá! Cleo Pires sai na frente para conseguir papel em filme de Stallone |access-date=2018-04-15 |archive-date=2016-03-04 |archive-url=https://web.archive.org/web/20160304031658/http://ego.globo.com/Gente/Noticias/0,,MUL964286-9798,00-QUASE+LA+CLEO+PIRES+SAI+NA+FRENTE+PARA+CONSEGUIR+PAPEL+EM+FILME+DE+STALLONE.html |url-status=dead }}&lt;/ref&gt; The role ended up with the actress [[Giselle Itié]].&lt;ref&gt;[http://cinema.uol.com.br/ultnot/2010/07/22/quando-o-stallone-disse-que-a-duble-faria-a-cena-de-tortura-eu-quase-chorei-diz-gisele-itie-sobre-os-mercenarios.jhtm &quot;Quando o Stallone disse que a dublê faria a cena de tortura, eu quase chorei&quot;, diz Giselle Itié sobre &quot;Os Mercenários&quot;]&lt;/ref&gt; Again, she was named one of the 25 sexiest women in the country, by ''Istoé Gente'' magazine .
[[John Candy]] was contractually mandated to make this film due to his existing contract with [[Carolco Pictures]] from their scrapped [[John Hughes (filmmaker)|John Hughes]] film ''Bartholomew v. Neff,'' which Candy was to star in with [[Sylvester Stallone]]. Despite his misgivings about the script and due to the fact that he owed more than $1 million due to his stake as a minority owner with the [[Toronto Argonauts]] football team, Candy agreed to make ''Wagons East'' in [[Durango|Durango, Mexico]].&lt;ref&gt;Crane, Robert, (2017). Crane: Sex, Celebrity, and My Father's Unsolved Murder&lt;/ref&gt;
In 2006, professional boxer Cornelius 'K9' Bundrage was a participant on ESPN's &quot;The Contender&quot; Season 2 boxing reality television show, with Hall of fame host [[Sugar Ray Leonard]] and [[Sylvester Stallone]] in Hollywood, California. During the show, Bundrage was viewed by millions of fans each week for fourteen weeks straight. He also was named &quot;fan favorite&quot; out of sixteen boxers from all over the United States. Bundrage was the last picked, winding up on the '''Blue Team'''. Wanting to avenge the slight, he volunteered to be the first member of his team to box in the tournament (quoting Matthew 20:16, &quot;And the last shall be first&quot;) and found himself in a match against [[Michael Clark (boxer)|Michael Clark]]. In a tightly fought battle for five rounds, the two found themselves practically tied on points, Bundrage knocked Clark down with a massive right hand to the temple, which won him the fifth round on 2 of the 3 judge's scorecards and won a majority decision (2 judges said Bundrage won, 1 said it was a draw). Bundrage moved along in the tournament, while Clark was forced to hang up his gloves and go home.
```



```
1335 She has worked extensively in the motion picture industry as a personal artist to [[Sharon Stone]], [[Lindsay Lohan]], [[Sylvester Stallone]], [[Gina Gershon]], [[Kate Beckinsale]], [[Brittany Murphy]], [[Michael Douglas]], [[Geena Davis]], [[Scarlett Johansson]] and [[Jessica Alba]]. She has also done several films as department head, along with countless print campaigns, media, awards, and television shows. Sawyer's work for TV series [[Mad Men]] was nominated for [[Emmy Award]] in 2011.&lt;ref&gt;{{Cite web ¦title=Tricia Sawyer ¦url=https://www.emmys.com/bios/tricia-sawyer ¦ website=Television Academy ¦publisher=[[Emmy Award]] ¦location=U.S. ¦access-date=2020-05-09 ¦archive-url=https://web.archive.org/web/20171001214244/http://www.emmys.com/bios/tricia-sawyer ¦archive-date=2017-10-01 ¦url-status=live }}&lt;/ref&gt;
1336 ¦Makeup artist: Sylvester Stallone
1337 ¦makeup artist: Sylvester Stallone and Gina Gershon
1338 ¦ Makeup artist for [[Sylvester Stallone]]
1339 Many of Kauk's exploits have been caught on film. In 1989 he climbed a five pitch route called ''Backbone'' (5.13a) at [[Smith Rock State Park¦Smith Rock]] for [[NBC]] Sportsworld,&lt;ref&gt;Alan Watts, ''Climber's Guide to Smith Rock'', Chockstone Press, 1992&lt;/ref&gt; and he repeated his toughest [[first ascents]] for Eric Perlman's ''Masters of Stone'' videos.  In 1992, Kauk trained and doubled [[Sylvester Stallone]] along with late climber [[Wolfgang Güllich]] for [[Cliffhanger (movie)¦Cliffhanger]]. He also doubled [[Michael Rooker]], [[Leon Robinson¦Leon]], and actress [[Janine Turner]]. In 1999 Kauk trained [[Tom Cruise]] for the climbing scenes in ''[[Mission: Impossible 2]]''.
1340 *[[Chuck Wepner]], a relatively unknown boxer, went up against world heavyweight champion [[Muhammad Ali]] in a bout in [[Cleveland]].  Wepner, a &quot;[[club fighter]]&quot; who had been selected as an easy opponent for the champ, knocked Ali down to the canvas in the ninth round, then went on to do what few of Ali's opponents had been able to do, &quot;going the distance&quot; for the full 15 rounds.  The Ali-Wepner fight was watched on closed circuit TV by an out of work actor, [[Sylvester Stallone]], who turned his own idea about an obscure boxer, getting a title shot, into the film ''[[Rocky]]'', with Stallone portraying the Wepner-like Rocky Balboa.&lt;ref&gt;George Foreman and Ken Abraham, ''Knockout Entrepreneur'' (Thomas Nelson Inc, Aug 11, 2009) pp81-82&lt;/ref&gt;
1341 The bulk of Lerner's films featured action stars who were at their peak in the 1980s-1990s: [[Jean-Claude Van Damme]], [[Steven Seagal]], [[Wesley Snipes]] and [[Dolph Lundgren]], usually released as [[direct-to-video]]. Lerner produced ''[[Rambo (2008 film)¦Rambo]]'' starring [[Sylvester Stallone]], and ''[[Righteous Kill]]'' starring [[Robert De Niro]] and [[Al Pacino]] for Millennium Films and [[Emmett/Furla/Oasis Films¦Emmett/Furla Films]].&lt;ref&gt;{{Cite web¦url=https://www.imdb.com/news/sb/2007-05-17 ¦title=De Niro, Pacino Team for Indie Movie ¦website=[[IMDb]] ¦url-status=dead ¦ archive-url=https://web.archive.org/web/20070520134535/http://imdb.com/news/sb/2007-05-17/ ¦archive-date=May 20, 2007 ¦df=mdy }}&lt;/ref&gt; He produced ''[[The Expendables (2010 film)¦The Expendables]]'', directed by Stallone.
1342 Starting in 1980, Long began working for David Frost Productions, writing and producing ABC and BBC specials, including the International Guinness Book of World Records, which also became a long-running syndicated series. Moving into feature films in the 1990s, Long worked on the second unit for dozens of motion pictures, including the [[Rambo (film series)¦Rambo]] series. His novella, ''Rogue's Babylon'', was the basis for Sylvester Stallone's hit movie, ''[[Cliffhanger (film)¦Cliffhanger]]''. During this time Long also worked for Bennett Productions, in Santa Monica, writing and producing action sport shows for TBS, NBC, CBS, FOX, HDTV, RUSH and others. Highlights include the Emmy nominated Red Bull Cliff Diving World Championships in La'nai, Hawaii, and the International Monitor Award (International Emmy) Winning show, ''Hawaiian Waterman'' (big wave surfing), for Asahi (Japanese HD TV).  Later work with New Wave Entertainment involved writing and producing long form shows for Discovery, History Channel, Showtime, A&amp;E, Lifetime, and HBO. His 2008 documentary for Code Black Entertainment and ''Ebony'' Magazine, ''Why We Laugh: The History of African American Humor'', was a featured documentary (Spectrum Series) at the 2009 Sundance Film Festival. His most recent (2010) DVD, ''Who You Callin' Crazy?'' featured comedian [[Katt Williams]].
1343 According to current [[United States]] [[copyright]], copyright owners have the [[exclusive right]] &quot;to prepare derivative works based upon [their] copyrighted work.&quot;&lt;ref&gt;[http://www.copyright.gov/title17/circ92.pdf 17 U.S.C. §106(2)(2009).] {{webarchive¦url=https://web.archive.org/web/20111009143055/http://www.copyright.gov/title17/circ92.pdf ¦ date=2011-10-09 }}&lt;/ref&gt; In the case where a copyright owner chooses to exercise their exclusive right to prepare derivative works against a work of fanfiction, they can sue the fanfiction writer for [[copyright infringement]]. To prove infringement, an owner must present evidence establishing that the accused has copied protected elements of the original work.&lt;ref&gt;{{cite web¦ url=http://chillingeffects.org/fanfic/faq.cgi#QID310¦title=Fan Fiction¦publisher=¦accessdate=8 February 2016}}&lt;/ref&gt;  If proven, possible infringement remedies include an order to cease sharing and/or to destroy the work (known as an injunction), or monetary damages.  The remedy is dependent on the harm done to the copyright owner, the intent of the infringing person, and the grievousness of the infringement.&lt;ref&gt;[http://www.copyright.gov/title17/circ92.pdf 17 U.S.C. §502 - 504(2009).] {{webarchive¦url=https://web.archive.org/web/20111009143055/http://www.copyright.gov/title17/circ92.pdf ¦date=2011-10-09 }}&lt;/ref&gt;   An example of injunction as remedy was seen in the case of ''[[Anderson v. Stallone]]''.  There, Sylvester Stallone successfully pursued an action for copyright infringement against Anderson, an author who wrote a proposed script for ''[[Rocky IV]],'' by proving that the copyright-protected characters used in the previous ''Rocky'' movies were central to the new script.  The court enjoined Anderson from pursuing the creation of a movie or other published work based on his script.&lt;ref&gt;{{cite web ¦ title=Anderson v. Stallone (11 U.S.P.Q.2D (BNA) 1161) ¦ website=Chicago-Kent College of Law ¦ date=26 April 1989 ¦ url=http://www.kentlaw.edu/faculty/rwarner/classes/legalaspects_ukraine/copyright/cases/anderson_v_stallone.html ¦ access-date=11 November 2018}}&lt;/ref&gt;
1344 ¦ full name          = Timothy Burton Anderson v. Sylvester Stallone, Freddie Fields, Dean Stolber, Frank Yablans, and MGM/UA
1345 '''''Anderson v. Stallone''''', 11 U.S.P.Q.2d 1161 ([[U.S. District Court for the Central District of California¦C.D. Cal.]] 1989) was a copyright infringement lawsuit against [[Sylvester Stallone]], [[MGM]], and other parties over a script for Stallone's film ''[[Rocky IV]]''.&lt;ref&gt;{{Cite book ¦last=Frankel ¦first=Susy ¦url=https://books.google.com/books?id=KDmjDwAAQBAJ ¦title=The Object and Purpose of Intellectual Property ¦date=2019 ¦publisher=Edward Elgar Publishing ¦isbn=978-1-78990-249-5 ¦pages=337-338 ¦language=en}}&lt;/ref&gt; This script written by Timothy Anderson was unsolicited and unauthorized, a key fact that led to a decision in favor of the defendants in the lower court and was later resolved in an out-of-court settlement during the pendency of plaintiff's appeal.&lt;ref&gt;{{Cite book ¦last1=Calboli ¦first1=Irene ¦url=https://books.google.com/books?id=0ZbwCAAAQBAJ ¦title=Diversity in Intellectual Property: Identities, Interests, and Intersections ¦ last2=Ragavan ¦first2=Srividhya ¦date=2015-05-28 ¦publisher=Cambridge University Press ¦isbn=978-1-107-06552-9 ¦pages=331 ¦language=en}}&lt;/ref&gt;
1346 [[Category:Sylvester Stallone]]
1347 ¦ [[Sylvester Stallone]]
1348 A famous use of a Tyrolean traverse in popular culture was in the opening scene of the 1993 [[Sylvester Stallone]] film ''[[Cliffhanger (film)¦Cliffhanger]]'', where a mountain rescue climber (played by Stallone) unsuccessfully attempts to transport a woman across a high Tyrolean traverse, only to have her fall to her death.&lt;ref&gt;{{Cite web¦title = Reading Eagle - Google News Archive Search¦ url = https://news.google.com/newspapers?nid=1955&amp;dat=19930611&amp;id=TpguAAAAIBAJ&amp;sjid=NaEFAAAAIBAJ&amp;pg=3573,5717476&amp;hl=en¦website = news.google.com¦accessdate = 2015-11-12}}&lt;/ref&gt; This scene was later spoofed in the [[Jim Carrey]] comedy film ''[[Ace Ventura: When Nature Calls]]''.
1349 * ''[[The Party at Kitty and Stud's]]'', the debut of [[Sylvester Stallone]]
1350 ¦The [[British Royal Family¦Royal Family]] going abroad • Dog lips • Plucky underdog films&lt;ref group=&quot;fn&quot; name=&quot;Plucky Underdog&quot;&gt;Such as ''[[Rocky]]'' and [[Escape to Victory]]. During this segment Ross claims that [[Sylvester Stallone]]'s character playing as a goalkeeper runs up the field with the ball and scores the winning goal, when, in fact, this never happens and Stallone instead saves a penalty resulting in the match being a draw&lt;/ref&gt;
1351 Other film gurus that signed included actors: [[Michael Douglas]], [[Dennis Hopper]], [[Sylvester Stallone]], [[Bruce Willis]], [[Danny DeVito]], [[Don Johnson]], [[James Woods]], [[Kelly Preston]], [[Patricia Heaton]] and [[William Hurt]]; and directors: [[Ridley Scott]], [[Tony Scott]], [[Michael Mann (director)¦Michael Mann]], Dick Donner and [[Sam Raimi]]; as well as star tennis player [[Serena Williams]] and media tycoon [[Rupert Murdoch]]. Supporters of the ad included chairman and owner of [[Paramount Pictures]], [[Sumner Redstone]], and billionaire mogul, [[Haim Saban]].
1352 [[Category:Images of Sylvester Stallone]]</text>
1353 * [[Sylvester Stallone]]
1354 '''''Shade''''' is a 2003 American [[neo-noir]] [[crime film¦crime]] [[thriller (genre)¦thriller film]] directed and written by Damian Nieman and starring [[Stuart Townsend]], [[Gabriel Byrne]], [[Thandiwe Newton]], [[Jamie Foxx]], [[Melanie Griffith]] and [[Sylvester Stallone]]. The film follows a trio of grifters who attempt to set up a legendary card shark nicknamed &quot;the Dean&quot;. The film had a limited release in the United States on May 7, 2004.
1355 * [[Sylvester Stallone]] as Dean &quot;The Dean&quot; Stevens
1356 ¦Based on movie ''[[Cliffhanger (film)¦Cliffhanger]]''. Novelization of the screenplay by Michael France and [[Sylvester Stallone]]
1357 In 1985, Tepper signed with [[Scotti Brothers Records¦Scotti Brothers]] and moved to Los Angeles. Actor/director [[Sylvester Stallone]] was taken with Tepper's song &quot;No Easy Way Out&quot;, which subsequently led to its inclusion in the [[film¦movie]] ''[[Rocky IV]]'' and to his song &quot;Angel of the City&quot; to be included in Stallone's 1986 film ''[[Cobra (1986 film)¦Cobra]]''. &quot;No Easy Way Out&quot; climbed into the Top 40, reaching #22 on Billboard's Hot 100 in 1986, and momentarily putting Tepper in the public eye. Several European labels, including Ministry of Sound, released &quot;No Easy Way Out&quot; with their own dance versions, resulting in regular airplay on German radio stations that continues today (in particular evening dance and party programming in Munich featured on Charivari 95.5 and Radio Gong 96.3). The heavy metal group [[Bullet For My Valentine]] also recorded the song in 2008. Tepper released two solo albums for Scotti Bros. Records, but both albums received little promotion from the label. In 1986, one year after ''Rocky IV'', another song from Tepper's debut album (titled ''[[No Easy Way Out (album)¦No Easy Way Out]]'') called &quot;Angel Of The City&quot; became the insert montage song of Stallone´s film ''[[Cobra (1986 film)¦Cobra]]''. Also in 1986, he co-wrote the single &quot;Le Bel Age&quot; for [[Pat Benatar]], which peaked at #54 on the Billboard charts. A second album, ''[[Modern Madness]]'' was released in 1988. He went on to join the classic hard rock group [[Iron Butterfly]] for a few years. His third solo album ''No Rest For The Wounded Heart'' was released exclusively in Europe in 1996 on the MTM Music label.
1358       <text bytes="15249" sha1="a9jwi3e800yqz3j9rqtaoilm9nal8nh" xml:space="preserve">{{Short description¦1978 film by Sylvester Stallone}}
1359 ¦ director       = [[Sylvester Stallone]]
1360 ¦ writer         = Sylvester Stallone
1361 * Sylvester Stallone
1362 '''''Paradise Alley''''' is a 1978 American [[sports film¦sports]] [[drama (film and television)¦drama film]] written, directed by, and starring [[Sylvester Stallone]] (in his feature directorial debut). The film tells the story of three [[Italian American]] brothers in [[Hell's Kitchen, Manhattan¦Hell's Kitchen]] in the 1940s who become involved in [[professional wrestling]]. [[Kevin Conway (actor)¦Kevin Conway]], [[Anne Archer]], [[Joe Spinell]], [[Armand Assante]], Lee Canalito, [[Frank McRae]], [[Joyce Ingalls]] and [[Tom Waits]] co-star in the film.
1363 * [[Sylvester Stallone]] as Cosmo Carboni
1364 Sylvester Stallone wrote the story as a novel then a screenplay before he wrote ''[[Rocky]]''. He later recalled:
1365 {{Sylvester Stallone}}
1366 [[Category:Films directed by Sylvester Stallone]]
1367 [[Category:Films with screenplays by Sylvester Stallone]]
1368 * 1946 - [[Sylvester Stallone]], American actor
```



1369 By the mid-1980s, Cannon was producing an average of 40 films per year and had a net worth of over US$1&amp;nbsp;billion.{{citation needed|date=May 2016}} Because of their fast, low-budget style of filmmaking, Globus and Golan earned the nickname &quot;the Go-Go Boys.&quot; Among the films produced by Cannon are ''[[Bloodsport (film)|Bloodsport]]'' with [[Jean-Claude Van Damme]], ''[[Superman IV: The Quest for Peace]]'' with [[Christopher Reeve]], ''[[King Lear (1987 film)|King Lear]]'' directed by [[Jean-Luc Godard]], ''[[Runaway Train (film)|Runaway Train]]'' (1985), ''[[Over the Top (1987 film)|Over The Top]]'' with [[Sylvester Stallone]], and ''[[Street Smart (film)|Street Smart]]'' with [[Morgan Freeman]]. During this time, Globus and Golan also acquired the rights to ''[[Spider-Man]]'' and ''[[Captain America]]''.{{citation needed|date=May 2016}}

1370 The screenplay for the film, written on [[Spec script|spec]] by Chris Soth as his thesis for the MFA Screenwriting program at USC, was originally purchased by the now defunct [[Savoy Pictures]]. When Savoy had the project, the film was going to be more epic in scale, with comprehensive visual and computer effects. Savoy even offered [[Sylvester Stallone]] $20 million to star, which he accepted. However, the studio went bust before the film was made. Twentieth Century Fox picked up the script from &quot;turnaround&quot; and fashioned it into a more intimate, smaller budgeted movie as they were looking to only spend $20 million on the picture. [[Graham Yost]] did an uncredited production polish on the script, which was also rewritten by four other writers on the way to production .&lt;ref&gt;{{Cite web|url=https://www.imdb.com/title/tt0120670/trivia?ref_=tt_ql_2|title = Firestorm (1998) - IMDb|website = [[IMDb]]}}&lt;/ref&gt;

1371 As early as the 1960s, the area was under development as &quot;a golf course and country club, named after [[Dean Martin]].&quot;&lt;ref name=wsjlasstarstuddedneighb&gt;{{cite news |last1=Jackson |first1=Candace |title=L.A.'s Star-Studded Neighborhood: Denzel Washington, Sylvester Stallone, Mark Wahlberg and other red-carpet regulars call this ritzy enclave home |url=https://www.wsj.com/articles/l-a-s-star-studded-neighborhood-1420736165 |accessdate=January 13, 2016 |newspaper=[[The Wall Street Journal]] |date=January 8, 2015}}&lt;/ref&gt; By 1979, developers Elliot Gottfurcht and Brian Adler, together with private investors, developed South Beverly Park as &quot;an idyllic community of historic-feeling grand estates that would feel like Beverly Hills of yesteryear.&quot;&lt;ref name=wsjlasstarstuddedneighb/&gt; The project was completed in 1990 as a 250-acre (100 ha) guard gated community.&lt;ref&gt;{{cite news |first=Mathis |last=Chazanox |url=https://www.latimes.com/archives/la-xpm-1989-11-12-me-2103-story.html |title=Developers Tailor $100 Book for Estate-of-the-Art Advertising : Beverly Park: Houses in an unfinished project below Mulholland Drive are ranked with elegant mansions |newspaper=Los Angeles Times |date=November 12, 1989}}&lt;/ref&gt;&lt;ref&gt;{{cite news |url=https://www.nytimes.com/1984/10/07/us/beverly-hills-developer-offers-land-and-dreams.html |title=Beverly Hills Developer Offers Land and Dreams |newspaper=[[The New York Times]] |date=October 7, 1984}}&lt;/ref&gt; Adler used the concept of having gates to distinguish the neighborhood from other [[Westside (Los Angeles County)|Westside]] neighborhoods of [[Beverly Hills, California|Beverly Hills]], [[Holmby Hills, Los Angeles|Holmby Hills]], and [[Bel Air, Los Angeles|Bel-Air]].&lt;ref name=LAT112008 /&gt; It originally contained 64 two-acre (0.8 ha) lots, a 4-acre (1.6 ha) landscaped park, and over 100 acres (40 ha) of open space; a number of adjacent lots have since been developed. [[Beverly Park]] is divided into two separate communities, one being North Beverly Park, the other South Beverly Park. North Beverly Park is a larger community and most homes there generally commanded much larger prices than homes of South Beverly Park.

1372 * [[Adele]]&lt;ref name=&quot;Flemming2022&quot;&gt;{{cite news |last=Flemming |first=Jack |date=February 25, 2022 |title=Adele buys Sylvester Stallone's Beverly Park mansion for $58 million - a deep discount |url=https://www.latimes.com/business/real-estate/story/2022-02-25/sylvester-stallone-sells-beverly-park-mansion-for-58-million-a-deep-discount |url-status=live |work=[[Los Angeles Times]] |archive-url=https://archive.today/20220619151432/https://www.latimes.com/business/real-estate/story/2022-02-25/sylvester-stallone-sells-beverly-park-mansion-for-58-million-a-deep-discount |archive-date=June 19, 2022 |access-date=June 20, 2022}}&lt;/ref&gt;

1373 * [[Sylvester Stallone]]&lt;ref name=Brennan/&gt;&lt;ref name=LAT112008 /&gt;&lt;ref&gt;{{Cite news|url=https://www.latimes.com/business/real-estate/story/2022-02-25/sylvester-stallone-sells-beverly-park-mansion-for-58-million-a-deep-discount|title=Sylvester Stallone sells Beverly Park mansion for $58 million|newspaper=Los Angeles Times|date=February 25, 2022|access-date=July 28, 2024}}&lt;/ref&gt;

1374 |ShortSummary=Snoop Dogg (Key) clears the air about his recent troubles with the law, then frames Jordan Peele for his crimes; three Muslim extremists (Barinholtz, Caeti, Key) fail in their plans to protest against America; a man's (McDonald) robot hands cause trouble in the office; a music video parody of &quot;SexyBack&quot; centered on Sylvester Stallone's comeback in the new ''Rocky'' movie; on ''Celebrity Pets'', Vince Vaughn's dog (voice of Keegan-Michael Key) reveals how Vaughn uses him to pick up women; Mel Gibson (McDonald) reveals revamped scenes for ''Apocalypto'' and thinly-veiled apologies for his anti-Semitic remarks; a parody for Domino's Pizza Fudge 'Ems features a disgusted pizza boy (Barinholtz) and a living turd visiting families; North Korea hires a loose-cannon scientist (Lee) to test missiles; Arden Myrin and Crista Flanagan interview celebrities at the Billboard Awards; 'Weekly News with Toby' centers on Michael Richards' racially-charged Laugh Factory rant and the aftermath; a newlywed couple (McDonald, Parker) struggle to tell the story of how they met; an animated piece featuring Kid Rock (voice of Ike Barinholtz) and Pamela Anderson's (voice of Crista Flanagan) wedding and divorce on the cover of a tabloid; a parody of ''Flavor of Love'' has Flava Flav and various animals as his potential girlfriends.

1375 ''Body Rock'' received negative reviews from critics and was a [[box office bomb|failure at the box office]]. Lamas' performance was nominated for [[Golden Raspberry Award for Worst Actor|Worst Actor]] at the [[5th Golden Raspberry Awards]], but lost to [[Sylvester Stallone]] in ''[[Rhinestone (film)|Rhinestone]]''. Also nominated was the song &quot;Smooth Talker&quot;-one of two songs in the film performed by Lamas. The other, &quot;Fools Like Me&quot;, remains Lamas' one single to date to crack the [[Billboard Hot 100|''Billboard'' Hot 100]] chart; it peaked at number 85 in January 1985.

1376 He followed that up with roles on ''[[Moonlighting (TV series)|Moonlighting]]'', ''[[Otherworld (TV series)|Otherworld]]'', ''[[Street Hawk]]'' and ''[[Knight Rider (1982 TV series)|Knight Rider]]'' before landing the role in the [[Sylvester Stallone]] vehicle, ''[[Cobra (1986 film)|Cobra]]'' (1986). Although the film was critically panned, it was a commercial success. ''[[The New York Times]]'' wrote of Thompson's portrayal, &quot;the archvillain, a character that is a cross between a [[James Bond]] fantasy villain such as Jaws and a raging psychopath, delivers a scorching monologue - a feat of linguistic sophistication that Cobra would have a hard time matching.&quot;&lt;ref name=NYTCobra&gt;{{cite news|last1=Darnton|first1=Nina|title=FILM: SYLVESTER STALLONE AS POLICEMAN, IN 'COBRA'|url=https://www.nytimes.com/movie/review?res=9A0DE5DF103AF937A15756C0A960948260|access-date=28 October 2014|work=New York Times|date=May 24, 1986}}&lt;/ref&gt;

1377 Kang played an [[Federal Bureau of Investigation|FBI]] agent in [[Jet Li]]'s film ''[[War (2007 film)|War]]'' (2007), and was featured in the movie ''[[Forbidden Warrior]]'' as Doran, a son of Genghis Khan. He had a small role in the action movie ''[[Live Free or Die Hard]]'', and he appeared in [[Walter Hill (filmmaker)|Walter Hill]]'s movie ''[[Bullet to the Head]]'' (2013), as Detective Taylor Kwon, opposite [[Sylvester Stallone]].

1378 '''Julia Nickson''' is a Singaporean-born American actress. She first came to the attention of audiences in the United States in the [[Sylvester Stallone]] film ''[[Rambo: First Blood Part II]]''. She appeared in the 2004 film ''[[Ethan Mao]]'' and in the 2008 independent film ''[[Half-Life (film)|Half-Life]]''.

1379 | Soundtrack to a [[Sylvester Stallone]] film. The song, a duet with longtime collaborator [[Bruce Roberts (singer)|Bruce Roberts]] was released as a single and also in a dance remix. A French version entitled &quot;Tant Qu'il Y Aura L'Amour&quot; was also issued and a Spanish version features on the Spanish DVD of the film.

1380 '''Andrea Kristen Savage''' (born February 20, 1973) is an American actress known best known for creating, writing and starring in the show ''[[I'm Sorry (TV series)|I'm Sorry]]'' (2017-2019). The show originally aired on [[TruTV]], but became a hit when it streamed on Netflix. In 2022, she starred as [[Stacy Beale]] opposite [[Sylvester Stallone]] in the series ''[[Tulsa King]]'' on [[Paramount+]]. That same year she also starred in ''Look Both Ways'' on Netflix and ''[[Beavis and Butt-Head Do the Universe]]'' for Paramount+. Savage is also known for roles in projects such as the [[Comedy Central]] mockumentary series ''[[Dog Bites Man]]'' (2006), the comedy film ''[[Step Brothers (film)|Step Brothers]]'' (2008), [[Hulu]]'s reality TV parody series ''[[The Hotwives]]'' (2014-2015), and the [[HBO]] comedy ''[[Veep (TV series)|Veep]]'' (2016-2017). She also has appeared in ''[[Curb Your Enthusiasm]]''.

1381 In 2022, she starred as [[Stacy Beale]], opposite [[Sylvester Stallone]] in the series ''[[Tulsa King]]'' on [[Paramount+]], which was quickly ordered for a second season.{{citation needed|date=March 2024}} That same year she also starred in ''Look Both Ways'' on Netflix and ''[[Beavis and Butt-Head Do the Universe]]'' for Paramount+.

1382 | Main role&lt;ref&gt;{{cite web |last=White |first=Peter |date=May 6, 2022 |title=''Tulsa King'': Andrea Savage To Star Alongside Sylvester Stallone In Taylor Sheridan's Paramount+ Mob Drama Series |url=https://deadline.com/2022/05/tulsa-king-andrea-savage-sylvester-stallone-taylor-sheridan-1235015650/ |website=[[Deadline Hollywood]]}}&lt;/ref&gt;

1383 [[Category:Images of Sylvester Stallone]]</text>

1384 |[[Bozo the Clown|Bozo's]] Client&lt;br /&gt;[[Sylvester Stallone|Sylvester Stallone's]] Assistant

1385 *2003: [[Fuck the Facts splits#Split with Sylvester Staline|Split with Sylvester Staline]] (Anvil of Fury Records)

1386 *2002: Winter 2002 Tour EP &lt;small&gt;(contains 4 songs taken from the upcoming split with [[Sylvester Staline]])&lt;/small&gt;

1387 Gracida was ranked 9 goals in [[England]], where he was reportedly HM [[Elizabeth II of the United Kingdom|Queen Elizabeth]]'s favorite player. He was a favorite instructor with celebrities and royalty and gave lessons to HRH [[Prince Charles]], HRH [[Prince William]], HRH [[Prince Harry]], [[Constantine II of Greece|King Constantine II of Greece]], HRH Prince Talal of [[Jordan]], [[James Packer]] and [[Sylvester Stallone]].&lt;ref name=Telegraph /&gt;

1388 Jonathan and Dio, the protagonist and antagonist, were created with the duality of light and shadow in mind, with the intention to have them contrast against each other: Dio was depicted as the embodiment of evil, and Jonathan as &quot;fundamentally virtuous&quot; and just. Araki also made use of black and white in his art to further contrast the two.&lt;ref&gt;{{cite book |last=Araki |first=Hirohiko |author-link=Hirohiko Araki |translator-last=Collins |translator-first=Nathan A. |date=2017 |title=Manga in Theory and Practice: The Craft of Creating Manga |publisher=[[Viz Media]] |page=56 |isbn=978-1-4215-9407-1}}&lt;/ref&gt; Jonathan was designed to be a symbol for the story and setting, something Araki would continue doing with later protagonists in the series.&lt;ref name=&quot;ann interview 2017-06&quot;&gt;{{cite web |url=http://www.animenewsnetwork.com/feature/2017-06-29/interview-jojo-bizarre-adventure-creator-hirohiko-araki/.118032 |title=Interview: Jojo's Bizarre Adventure Creator Hirohiko Araki |last1=Mitchem |first1=Casey Lee |last2=Silverman |first2=Rebecca |date=2017-06-29 |website=[[Anime News Network]] |access-date=2017-09-30 |archive-url=https://web.archive.org/web/20170708115039/http://www.animenewsnetwork.com/feature/2017-06-29/interview-jojo-bizarre-adventure-creator-hirohiko-araki/.118032 |archive-date=2017-07-08 |url-status=live}}&lt;/ref&gt; Because Araki wanted the series to move on with new characters in the Joestar family, Jonathan was specifically written as &quot;the first Joestar&quot; that would function as a symbol of purity and dignity, and not as a unique character. This limited what Jonathan could do; in retrospect, Araki considered him passive and &quot;a bit boring&quot;.{{citation needed|date=December 2020}} Jonathan's physical transformation during the seven year skip was done with his upcoming battle with Dio in mind and inspired by muscular film actors popular at the time, such as [[Arnold Schwarzenegger]] and [[Sylvester Stallone]]. A fan of karate manga as a child, Araki also wanted JoJo to exude an aura of strength like in ''[[Karate Baka Ichidai]]'', leading to the character learning the supernatural Hamon technique.&lt;ref&gt;{{cite book |author=Araki, Hirohiko|title=JoJo's Bizarre Adventure: Part 1 Phantom Blood|volume=1|page=249|date=September 2, 2014|publisher=[[Viz Media]]|isbn=978-1-4215-7879-8}}&lt;/ref&gt;

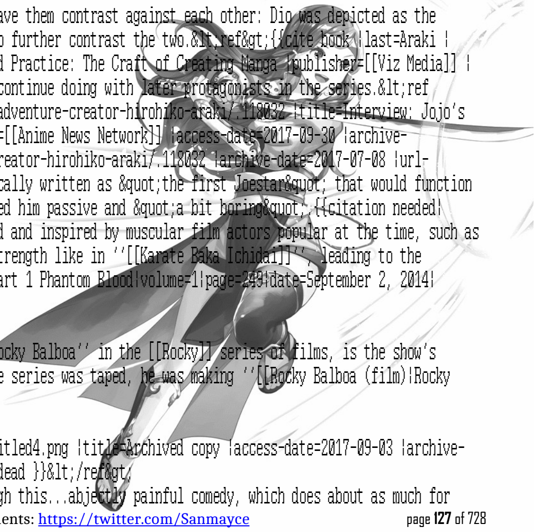
1389 |caption = North American SG cover artwork featuring [[Sylvester Stallone]] as [[John Rambo]]

1390 The program was hosted by [[Sugar Ray Leonard]], former champion boxer. [[Sylvester Stallone]], who famously portrayed the boxer ''Rocky Balboa'' in the [[Rocky]] series of films, is the show's executive producer. It was also produced by [[Mark Burnett]]. Unlike in season 1, Stallone does not appear on camera; at the time the series was taped, he was making ''[[Rocky Balboa (film)|Rocky Balboa]]'', the sixth installment in the ''[[Rocky]]'' movie series. The movie premiered in December 2006.

1391 [[Category:Images of Sylvester Stallone]]</text>

1392 [[Sylvester Stallone]] appeared as an extra in the restaurant scene.&lt;ref&gt;{{Cite web |url=https://s12.postimg.org/4fdyk94ul/Untitled4.png |title=Archived copy |access-date=2017-09-03 |archive-url=https://web.archive.org/web/20170903212447/https://s12.postimg.org/4fdyk94ul/Untitled4.png |archive-date=2017-09-03 |url-status=dead }}&lt;/ref&gt;

1393 Film critic and historian [[Leonard Maltin]] seemed to agree. &quot;A [[hemorrhoid]]-and-a-half to anyone who sits all the way through this...abjectly painful comedy, which does about as much for



Reynolds' career as ''[[Stop! Or My Mom Will Shoot]]'' did for [[Sylvester Stallone]]'s. That it was Ray Sharkey's last movie adds insult to injury.&quot;&lt;ref&gt;{{cite book|title=Leonard Maltin's 2009 Movie Guide|last=Maltin|first=Leonard|publisher=[[Penguin Group]]|year=2008|isbn=9780452289789|page=281|url=https://books.google.com/books?id=6EgPDierNGUC&amp;pg=PA281}}&lt;/ref&gt;

1394 Araki called Joseph a swindler in comparison to the gentlemanly Jonathan, because he is constantly looking to win and will engage in reckless behavior without hesitation. This was not only to create contrast between the two, but also because the author wanted to shift from the physical battles in Part 1 to more &quot;cerebral&quot; fights. He wanted Joseph to be a ''[[shōnen manga]]'' hero who bends the rules as he fights, like the protagonist of his earlier series ''Cool Shock B.T.'', winning by cunning and logic, instead of bravery and perseverance. Araki fleshed the character's personality out by having him predict the actions of his opponents ahead of time. He referred to Joseph as &quot;a muscle-bound B.T.&quot;, achieved by adding some [[Sylvester Stallone]] to B.T., plus some added cheerfulness to make him &quot;jolly.&quot; Araki stated that Joseph is the character that connects the Joestar bloodline to Parts 3 and 4, and although he had Jonathan die in Part 1, he never even considered killing off Joseph.&lt;ref name=&quot;joseph&quot;/&gt;

1395 That same year Richards directed [[Robert Mitchum]] and [[Charlotte Rampling]] in the [[Raymond Chandler]] adaptation, ''[[Farewell, My Lovely (1975 film)|Farewell, My Lovely]]'' (1975). [[Roger Ebert]] said the movie &quot;Never steps wrong&quot; and called it &quot;a totally assured piece of work&quot;.&lt;ref&gt;Ebert, Roger &quot;[http://www.rogerebert.com/reviews/farewell-my-lovely-1975 Reviews: Farewell, My Lovely]&quot; [[RogerEbert.com]] January 1, 1975&lt;/ref&gt; [[Sylvia Miles]] earned a Best Supporting Actress Nomination. [[Sylvester Stallone]] also appears on screen in an early, pre-''[[Rocky]]'' role. Film critic [[Rex Reed]] said, &quot;''Farewell, My Lovely'' is the kind of movie [[Humphrey Bogart]] would have stood in line to see.&quot;&lt;ref&gt;Townsend, Guy M.&quot;[https://books.google.com/books?id=-9koH5oNBvQC&amp;q=rex+reed+farewell+my+lovely+bogart&amp;pg=PA9 The Mystery Fancier (Vol. 1 No. 6) November 1977] Page 9&quot;&lt;/ref&gt;

1396 And just incase you were wondering where Zachary gets his good looks from, well you will never know. His mystery is only exceeded by his power. Which he has alot of because he shares genes with Sylvester Stalone.

1397 Florida is also the home of many famous actors. Celebrity residents of Florida have included [[Burt Reynolds]], [[Sylvester Stallone]], [[John Travolta]], [[Kelly Preston]], [[Andy Garcia]], [[Tom Cruise]], [[Ron Palillo]], [[Kaye Stevens]], [[Michael Winslow]], [[Larry Haines]], [[Margaret Hayes]], [[Roberta Haynes]], [[Donald Sutherland]], [[DJ Khaled]], [[Meinhardt Raabe]], [[Jackie Gerlich]], [[Steven Bauer]], [[Manolo Villaverde]], [[Hedy Lamarr]], [[Susana Giménez]], [[Liv Ullman]], [[Geraldine Chaplin]], [[Sydney Sweeney]], [[David Caruso]], [[Kenan Thompson]], [[Vanilla Ice]], [[Christian Daniel]], [[Howard Stern]], [[Scott Baio]], [[Susan Buckner]], [[Iggy Pop]], [[Donna Summer]], [[George Keymas]], [[Nicolas Coster]], [[Margaret Kerry]], [[Antonio Sabàto Jr.]], [[Gloria Estefan]], [[Pitbull (rapper)|Pitbull]], [[Jackie Gleason]], [[Cindy Morgan]], [[Richard Boone]], [[Perry Como]], [[David Winters (choreographer)|David Winters]], [[Herb Shriner]], [[Jean Shepherd]], [[Bob McFadden]], [[Carrot Top|Scott &quot;Carrot Top&quot; Thompson]], [[Billy Connolly]], [[Cody Burger]], [[Mickey Kuhn]], [[Bobby Breen]], [[Peter Palmer (actor)|Peter Palmer]], [[Ricou Browning]], [[Gabrielle Anwar]], [[Philip Michael Thomas]], [[Eugene Levy]], [[Barrie Ingham]], [[Mike Douglas]], [[Leslie Nielsen]], [[Eddie Egan]], [[Kirby Grant]], [[Patricia Dane]], [[Norma Miller]], [[Lili Damita]], [[Martha O'Driscoll]], [[Dixie Dunbar]], [[June Preisser]], [[Phyllis Thaxter]], [[Vic Damone]], [[Vaughn Monroe]], [[Tom Fadden]], [[Kimbo Slice]], [[Rhonda Shear]], [[Luke Halpin]], [[Jay North]], [[Dan White (actor)|Dan White]], [[Dennis Hoey]], [[Freddie Bartholomew]], [[George Hamilton (actor)|George Hamilton]], [[Gary Burghoff]], [[Jo Ann Pflug]], [[Steve Sipek]], [[Arthur Metrano]], [[Dave Madden]], [[David Cassidy]], [[Davy Jones (musician)|Davy Jones]], [[Anthony Newley]], [[Gayle Hunnicutt]], [[Cindy Crawford]], [[Gail Edwards]], [[Enrique Iglesias]], [[Sofia Vergara]], [[Hulk Hogan]], [[Nick Hogan]], [[Randy Savage]], [[Dave Bautista]], [[John Cena]], and [[Dwayne Johnson]]. There are also quite a few successful film directors located in the state including [[Daniel Myrick]], [[George Abbott]], and [[Harry Revier]].

1398 | Mother of [[Sylvester Stallone]]

1399 '''Brigitte Nielsen''' (born 15 July 1963) is a Danish actress, who entered the house last on Day 1. She was once married to [[Sylvester Stallone]]. During her time in the house she became close to Lisa and Caprice. She was known for her use of her catchphrase &quot;diggity&quot;.

1400 '''Jackie Stallone''' (29 November 1921 – 21 September 2020) was an American astrologer and mother to film actor [[Sylvester Stallone]]. She entered the house on Day 5, as part of the Mediaeval Task, in which she was referred to as &quot;the Queen mother&quot;. She is remembered for her first words to Brigitte upon entering the house, &quot;Yeah, Brackie&quot;. She is also former mother-in-law to fellow housemate Brigitte. Her entering upset Brigitte greatly and resulted in Brigitte requesting to leave. However, in the house, they resolved their issues with each other. She was the cause of much conflict between her fellow housemates particularly with Lisa who likened her to the [[Bride of Frankenstein]] and told her she hoped that she choked on her eggs. She also had many confrontations with John who branded her &quot;lazy&quot;. Her lack of day-to-day knowledge was much to the entertainment of her fellow housemates such as her not being able to open a bottle of wine or make tea. Jackie said that this was down to the fact that she had employees to do these things for her in the outside. After only just four days in the house, she was first to be evicted on Day 9, losing out to John, with 33% of the vote to save. She is the oldest housemate to ever enter the house in the UK, until 10 series later, when [[Lionel Blair]] entered the house at age 85.

1401 The cartoon generated a mild controversy at the production studio, with writers wondering how they could present a child-friendly main character who was created as a troubled [[Vietnam War]] veteran suffering from [[posttraumatic stress disorder]] (PTSD). While some have alleged that the show used child psychologists who recommended that the cartoon not make any references to Vietnam, POWs, or Rambo's experiences in 1982's ''First Blood'' and 1985's ''Rambo: First Blood Part II'', the show's head writer, Michael Chain, denies this and instead states that his previous experience in producing children's programming led him to make sure that Rambo would not &quot;have an adverse effect on the psyche of children&quot;.&lt;ref&gt;{{cite web |last1=Chernov |first1=Matthew |title=10 Secrets From the Cast &amp; Crew of the '80s Rambo Cartoon |url=https://www.toplessrobot.com/2015/05/rambo_force_of_freedom_stallone_first_blood_morrel.php |website=Topless Robot |date=4 May 2015 |access-date=31 August 2019}}&lt;/ref&gt; Sylvester Stallone was annoyed and embarrassed that the Rambo character was used as a cartoon character.&lt;ref&gt;{{Cite web|url=https://www.chicagotribune.com/news/ct-xpm-1986-05-18-8602040747-story.html|title=Stallon wields a pen with Ramboian Power|website=Chicago Tribune|date=18 May 1986}}&lt;/ref&gt;

1402 [[Category:Cultural depictions of Sylvester Stallone]]

1403 | 1986 || ''[[Cobra (1986 film)|Cobra]]'' || Dr. Demopoulos || with [[Sylvester Stallone]]

1404 * ''[[The Expendables (2010 film)|The Expendables]]'' – 2010 film by [[Sylvester Stallone]] about a multinational military force tasked with assassinating a South American dictator

1405 *[[Rocky Balboa]], from the [[Rocky]] film series starring [[Sylvester Stallone]] as the films' title character, who is sometimes called Rocco.

1406 Smith returned to the screen in 1975 with a supporting role in the film adaptation of [[Raymond Chandler]]'s novel ''[[Farewell, My Lovely (1975 film)|Farewell, My Lovely]]'', starring [[Robert Mitchum]] as [[Philip Marlowe|Phillip Marlowe]], in which Smith was shown in a violent bedroom scene opposite a then near-unknown [[Sylvester Stallone]]. The year of 1976 saw the release of six films where she had minor or supporting roles. One of those films, Richard Lerner's ''[[Revenge of the Cheerleaders]]'' had been shot in 1974 when Smith was eight months pregnant. Also included in her 1976 releases are the [[Blaxploitation|blaxpoitation]] classic ''[[Drum (1976 film)|Drum]]'', [[Rene Daalder]]'s cult classic ''[[Massacre at Central High]]'', and ''[[Slumber Party '57]]''.

1407 * Dynamite Joe (voiced by [[Jeff Bennett]] impersonating [[Sylvester Stallone]]) – An explosives expert who disguises himself as Santa Claus for a crime spree.

1408 | ShortSummary = [[Sylvester Stallone|Sly]] [[Arnold Schwarzenegger|Eastenegger]], a famous action movie star, and his [[Steven Spielberg|director]] choose Edge City to film his latest movie, and challenge The Mask over who will be the next action star. Meanwhile, Peggy discovers a stolen [[Nuclear weapon|nuclear bomb]] that Eastenegger is plotting to use in the film.

1409 Laure is also a film actress, appearing in a number of Canadian-produced films, including the controversial 1974 release by [[Dušan Makavejev]] ''[[Sweet Movie]]'', which was notable for both its sexual explicitness and [[scatology]]. Laure and Furey were frequent co-stars in the films of Gilles Carle, most notably, ''L'Ange et la femme'' (1977) and ''Fantastica'' (1980). She also stars alongside [[Pelé]], [[Sylvester Stallone]] and [[Michael Caine]] in the 1981 film ''[[Escape to Victory]]''.

1410 He has a strange style of introducing himself. He crosses his arms, keeping them at a small distance from his body, poses his head at an angle to his torso, and says this English-language sentence in the local accent: &quot;My name is Rambo Rambo, John Rambo, Silver Stallone, Cockroach Killer.&quot;&lt;ref name=Tribune/&gt; He always mispronounces &quot;[[Sylvester Stallone]]&quot; as &quot;Silver Stallone.&quot; When he is excited, he adds &quot;After Kar&quot; at the beginning of his signature sentence. This is a combination of &quot;After All&quot; and &quot;Aakhir Kar&quot; (Urdu: آخر کار). Both of these phrases have the same meanings in English and Urdu, respectively.

1411 :Well, she's not particularly mainstream, but what about [[Traci Lords]]? And there is that [[Sylvester Stallone]] movie... [[User:Adam Bishop|Adam Bishop]] 20:33, 27 September 2006 (UTC)

1412 |[[Sylvester Stallone]]

1413 *[[Rambo (franchise)|''Rambo'' (franchise)]], starring Sylvester Stallone, commonly named after the main character

1414 As a lead actress, Fluegel starred in the independent film ''[[Freeway (1988 film)|Freeway]]'' (1988). In 1989, she starred in [[Lock Up (1989 film)|''Lock Up'']] with Sylvester Stallone. She later had a recurring role on ''[[Wiseguy (TV series)|Wiseguy]]''.&lt;ref name=bio/&gt;

1415 West would eventually work on a number of action films including the 2012 film ''[[The Expendables 2]]'' with [[Sylvester Stallone]], [[Jet Li]], [[Chuck Norris]], [[Jean-Claude Van Damme]], [[Bruce Willis]], and [[Arnold Schwarzenegger]].&lt;ref name=2013interview/&gt; He also reunited with Nicolas Cage in 2012 for ''[[Stolen (2012 film)|Stolen]]''.&lt;ref name=2013interview/&gt; In 2015, he directed the crime thriller film ''[[Wild Card (2015 film)|Wild Card]]'' with [[Jason Statham]]. It is a remake of the 1986 film ''[[Heat (1986 film)|Heat]],'' based on [[Heat (William Goldman novel)|the novel of the same name]] by [[William Goldman]].&lt;ref name=2013interview&gt;{{cite web |url=http://denofgeek.com/movies/stolen/26723/director-simon-west-stolen-con-air-rick-astley-and-more1veebmhm318wuxyfo7b2vmpwo |title=Director Simon West: Stolen, Con Air, Rick Astley and more |access-date=23 November 2019 |last=Lambie |first=Ryan |date=5 August 2013 |publisher=Den of Geek}}&lt;/ref&gt; In 2017 West directed his first feature musical comedy ''[[Gun Shy (2017 film)|Gun Shy]]'' starring [[Antonio Banderas]] and based on the novel ''Salty'' by Mark Haskbell Smith. {{cn|date=April 2024}} West directed the 2019 big-budget disaster movie from China, ''[[Skyfire (film)|Skyfire]]'',&lt;ref name=lh/&gt;&lt;ref name=var0514&gt;{{cite web |url=https://variety.com/2019/film/news/cannes-skyfire-simon-west-meridian-base-fx-1203214530/ |title=Cannes: 'Skyfire,' China's First Big-Budget Disaster Movie, Shopped by Meridian Ent. (EXCLUSIVE) |access-date=14 May 2019 |last=Davis |first=Rebecca |date=14 May 2019 |work= [[Variety (magazine)|Variety]]}}&lt;/ref&gt; and will also direct another Chinese action film, ''[[The Legend Hunters]]''.&lt;ref name=lh&gt;{{cite web |url=https://variety.com/2019/film/news/why-simon-west-making-movies-in-china-1203244782/ |title=Why Simon West Is Making Movies in China (EXCLUSIVE) |access-date=23 November 2019 |last=Davis |first=Rebecca |date=15 June 2019 |work=Variety}}&lt;/ref&gt;

1416 In films, he played Dozer, a biker with a severe speech impediment who befriends [[Roy L. Dennis|Rocky Dennis]] and his mother in the movie ''[[Mask (1985 film)|Mask]]''. Burkley appeared in the drug culture drama ''[[Rush (1991 film)|Rush]]'' in 1991, the [[Chevy Chase]] comedy ''[[Fletch Lives]]'', and [[Sylvester Stallone]]'s ''[[Stop! Or My Mom Will Shoot]]''. He appeared in the [[Kevin Costner]] golf comedy ''[[Tin Cup]]''. Other movies included ''[[The Doors (film)|The Doors]]'' and ''[[Murphy's Romance]]''.

1417 In 1984, Torien formed the band Touch with multi-instrumentalist [[Marc Danzeisen]], future [[Icon (band)|Icon]] frontman Jerry Harrison, and future [[Chalk FarM]] bassist Orlando Sims. The group recorded a five-song demo, which was heard by [[Sylvester Stallone]] during his search for new acts to perform songs for [[Rocky IV]]. The band recorded &quot;The Sweetest Victory&quot; which did not appear in the film but was included on the soundtrack. The song was featured in Stallone's director's cut of the film in 2021.

1418 Kael reviews 117 films in this eighth collection. She gives rich praise to the work of performers and directors she admires, for example, [[Molly Ringwald]]'s performance in ''[[Sixteen Candles]]'', – [[Steve Martin]], [[Robin Williams]], and [[Nick Nolte]], (three perennial Kael favourites), and [[Luchino Visconti]], for his work on [[The Leopard (1963 film)|The Leopard]] – &quot;''The Leopard'' is so beautifully felt that it calls up a whole culture. It casts an intelligent spell – intelligent and rapturous.&quot; (Though Visconti's film had originally been released in 1963, Kael is here reviewing the release of a full length three hours and five minutes version). And she's typically cool to work she regards as second rate; 'When you come out of [[Desperately Seeking Susan]], you don't want to know who the director is – you want to know who the perpetrator is.' And [[Steven Spielberg]]'s segment of [[Twilight Zone: The Movie]] – a, &quot; lump of ironclad whimsy. It's as if Steven Spielberg had sat down and thought out what he could do that would make his detractors happiest.&quot; And [[Sylvester Stallone]]'s work on ''[[Staying Alive (1983 film)|Staying Alive]]'': &quot;What can be done about this mock writer-director-producer-actor? He has become the stupidos' [[Orson Welles]].&quot;

1419 *[[September 6]]: The Philadelphia Art Commission approves the placement of a statue of [[Sylvester Stallone|'''Sylvester Stallone's film character, Rocky Balboa,''']] from the film ''[[Rocky III]]'' near the [[Rocky Steps|steps]] of the [[Philadelphia Museum of Art]]. [http://www.nbc10.com/news/9796121/detail.html (WCAU)]

1420 [[Category:Cultural depictions of Sylvester Stallone]]</text>

1421 '''''First Blood''''' is a 1972 American [[Action fiction|action]]-[[Thriller (genre)|thriller]] novel by [[David Morrell]] about a troubled [[Homelessness in the United States|homeless]] [[Vietnam War]] veteran, known only by his last name of [[John Rambo|Rambo]], who wages a brutal one-man war against local and state police in [[Kentucky]]. It was adapted into the 1982 film ''[[First Blood]]'' starring [[Sylvester Stallone]],&lt;ref&gt;{{cite web |url=http://davidmorrell.net/books/first-blood/ |title=Rambo |access-date=July 21, 2012 |author=David Morrell|author-link=David Morrell }}&lt;/ref&gt; which ended up spawning an entire [[Rambo (franchise)|media franchise]] around Rambo.

1422 In 1972, Morrell sold the film rights to ''First Blood'' to [[Columbia Pictures]], who in turn sold them to [[Warner Bros.]] The film languished in [[development hell]] for ten years, with the story passing through three companies and eighteen screenplays. Finally, [[Andrew G. Vajna]] and [[Mario Kassar]], two film distributors looking to become producers, obtained the film rights. [[Sylvester Stallone]] was cast in the lead role, due to the star power he had from the films ''[[Rocky (film)|Rocky]]'' and ''[[Rocky II]]''. Stallone used his clout to force changes to the script to make Rambo a more sympathetic character, including having Rambo not directly kill any police or national guardsmen (in the novel, he kills many), and having him survive at the end instead of dying as he does in the book. The novel also differs from the film in that, instead of Trautman personally training and leading Rambo in Vietnam, Trautman is mentioned as the commander of the school where he trained and Rambo barely remembers him. Also in the book, Rambo expresses anger that Trautman would turn on one of his own after hearing that Trautman is working with the police to capture him.

1423 In 1991, he travelled to Los Angeles and secured his first Hollywood acting role in the 1993 action blockbuster ''[[Cliffhanger (film)|Cliffhanger]]'' with [[Sylvester Stallone]].&lt;ref name=&quot;tv&quot;&gt;{{Cite web |title= Craig Fairbrass credits |url= https://www.tvguide.com/celebrities/craig-fairbrass/credits/3030064420/ |work= tvguide.com |access-date=31 May 2023}}&lt;/ref&gt; This led to roles in independent films with lead roles in ''[[Beyond Bedlam]]'' (1994),&lt;ref name=&quot;tv&quot;/&gt; with [[Elizabeth Hurley]], ''[[Galaxis]]'' (1995),&lt;ref name=&quot;tv&quot;/&gt; with [[Brigitte Nielsen]], ''Proteus'', ''Killing Time'', and ''Darklands''.&lt;ref name=&quot;tv&quot;/&gt;

1424 He rose to fame by playing a janitor nicknamed ''[[John Rambo|Jan Rambo]]'' in [[Pakistan Television Corporation|PTV]]'s famous comedy show ''[[Guest House (TV series)|Guest House]]'' (1991). He later assumed ''Jan Rambo'' as his [[stage name|screen name]] after the [[Sylvester Stallone]] character [[John Rambo]]. People on the TV stage had nicknamed him ''Jan Rambo'' due to his face, though certainly not his physique, resembling the Hollywood actor's.&lt;ref name=&quot;Tribune&quot;&gt;{{cite news | url=https://tribune.com.pk/story/226934/afzal-khan-the-pakistani-rambo/ | title=Afzal Khan: The Pakistani 'Rambo' | newspaper=The Express Tribune (newspaper)|date=8 August 2011 | first=Saadia | last=Qamar | accessdate=14 June 2020}}&lt;/ref&gt;&lt;ref name=&quot;tv.com.pk&quot; /&gt;

1425 Lipton previously had a slogan of long standing, &quot;the brisk tea&quot;.&lt;ref&gt;{{cite web | url=https://www.worthpoint.com/worthopedia/nos-vtg-40s-lipton-tea-box-contents-1981956628 | title=NOS VTG 40's LIPTON TEA BOX WITH CONTENTS NEW OLD STOCK 1944 ORANGE PEKOE BLACK &amp;#124; #1981956628 }}&lt;/ref&gt; Brisk is well known for its high-profile “That's Brisk, baby!” campaigns. The [[J. Walter Thompson]] ad agency first launched the campaign in 1996, featuring pop-culture icons in [[Clay animation|claymation]], and was revived in 2010 by creative agency Mekanism. The advertisements highlight exhausted celebrities who are dramatically reinvigorated by drinking Brisk. [[Frank Sinatra]] (voiced by [[Joe Piscapo]]), [[Babe Ruth]], [[Reggie Jackson]], [[Elvis Presley]], [[James Brown]], [[Coolio]], [[Willie Nelson]], [[Bruce Lee]], [[Danny DeVito]], [[Bruce Willis]], [[Ozzy Osbourne]], [[Sylvester Stallone]] (who voiced himself as [[Rocky Balboa]]), [[Danny Trejo]], [[Eminem]], and [[Doja Cat]] have all been featured in Brisk spots.&lt;ref&gt;{{cite web|title=Agency News|url=http://www.adweek.com/news/advertising/agency-news-37835| publisher=Adweek}}&lt;/ref&gt;

1426 &lt;li&gt;19:01, 15 Oct 2003 [[User:Jimfbleak|Jimfbleak]] deleted &quot;Silvester Stallone&quot; &lt;em&gt;(content was: 'hey')&lt;/em&gt;&lt;/li&gt;

1427 *In ''[[The Muppet Show]]'' episode #350, [[Sylvester Stallone]] sings a modified version of the song while dressed as a Roman gladiator and battling a dancing lion.

1428 '''''[[Rocky]]''''' is a 1976 [[motion picture]] written by and starring [[Sylvester Stallone]] as an [[Underdog (competition)|underdog]] [[boxing|boxer]]. It tells the rags-to-riches [[American Dream]] story of [[Italian-American]] &quot;[[Rocky Balboa (character)|Rocky Balboa]]&quot;, a slightly dull-witted but good-hearted &quot;collection agent&quot; for a [[loan shark]] in Philadelphia with a penchant for boxing, who gets a shot at the world heavyweight title. It was directed by [[John G. Avildsen]] who won an Oscar for his direction. The final fight scene in the movie was inspired by a real-life fight between [[Muhammad Ali]] and [[Chuck Wepner]]. Made for a budget of only $1.1 million dollars and shot in a record 28 days the film went on to make over US $117.2 million at the box office and spawn five sequels.</text>

1429 * 2016 : [[Sylvester Stallone]]

1430 The series is noted for popularizing [[Mr. T]], who won the &quot;America's Best [[Bouncer (doorman)|Bouncer]]&quot; (sometimes reported as &quot;World's Toughest Bouncer&quot;)&lt;ref name=&quot;Otfinoski2010&quot; /&gt; competition twice on the show, donating his $3000 prize to charity.&lt;ref name=&quot;Jet1981&quot;&gt;{{cite journal |title=Is Your Body Worth $500 A Day? This Man Will Guard It |journal=[[Jet (magazine)|Jet]] |date=January 15, 1981 |volume=59 |issue=18 |url=https://books.google.com/books?id=7LYDAAAAMBAJ&amp;pg=PA60 |accessdate=4 February 2019 | publisher=[[Johnson Publishing Company]] |issn=0021-5996 |via=[[Google Books]]}}&lt;/ref&gt; Mr. T was subsequently cast by [[Sylvester Stallone]] as [[Clubber Lang]], the villain in ''[[Rocky III]]''.&lt;ref name=&quot;Otfinoski2010&quot;&gt;{{cite book |last1=Otfinoski |first1=Steven |authorlink1=Steven Otfinoski |title=African Americans in the Performing Arts |date=2010 | publisher=[[Infobase Publishing]] |isbn=9781438128559 |page=164 |edition=Revised |chapter-url=https://books.google.com/books?id=gnXQSqTx2h0C&amp;pg=PT172 |accessdate=4 February 2019 |lccn=2009012400 |chapter=Mr. T |via=[[Google Books]]}}&lt;/ref&gt;&lt;ref name=&quot;McKenna2015&quot; /&gt;

1431 | Former financial adviser for [[Al Pacino]], [[Martin Scorsese]] and [[Sylvester Stallone]], pleaded guilty in 2010 to [[fraud]] and [[money laundering]] for diverting $33 million of his clients' money to pay for personal expenses.&lt;ref&gt;{{cite news|url=https://www.nytimes.com/2010/09/11/business/11fraud.html|title=Guilty Plea in Fraud by Adviser to Stars|newspaper=[[The New York Times]]| access-date=October 16, 2013 }}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.huffingtonpost.com/2011/03/03/ken-starr-hollywoods-mado_n_830918.html |title=Ken Starr, Hollywood's Madoff, Gets Sentenced |website=Huffingtonpost.com |date=March 3, 2011 |access-date=October 14, 2013 }}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.cnbc.com/id/45554694 |title=American Greed: Celebrity Scam Artist |work=CNBC |date=January 24, 2012 |access-date=October 14, 2013 }}&lt;/ref&gt;

1432 | [[Sylvester Stallone]]

1433 | Sylvester Stallone

1434 | Sylvester Stallone

1435 |Aux3 = [[Arnold Schwarzenegger]] vs. '''[[Sylvester Stallone]]'''

1436 ''Fight 3'': In a bizarre twist, both fighters punch and knock each other out, prompting referee Mills Lane to count them out. Arnold Schwarzenegger manages to stand up, but Sylvester Stallone sweep-kicks him to the floor and then stands up himself. Mills then declares Sylvester the winner. Arnold Schwarzenegger flicks a cigar into the audience that lands in Don King's hair, which sets him on fire and eventually burns him alive. Also featuring [[Adrian Zmed]].

1437 ''Fight 2'': Both fighters are given the chance to choose a setting of how the ring would look for the fight by a coin toss. John wants the set of ''[[Entertainment Tonight]]'', while David wants a ''[[Baywatch]]''-like setting. In the end, Hasselhoff wins and gets a ring set up to look like a beach. John Tesh is crushed by [[KITT]] when David Hasselhoff asks him for help after his face gets torn off. David Hasselhoff's face and skull are ripped off when John Tesh unfolds an umbrella in his head. He collapses on top of KITT after the fight. Also featuring [[Sylvester Stallone]], [[Arnold Schwarzenegger]], [[David Spade]] and [[Steven Seagal]].

1438 La Botz began his acting career after catching the attention of actor, director, and [[independent film]]maker [[Steve Buscemi]], who cast La Botz as a blues-singing convict in his 2000 film [[Animal Factory]]. La Botz performed two original songs in the film, &quot;This Ain't the Way I Came Up&quot; and &quot;Lay Down the Bottle.&quot;&lt;ref name=animal-factory/&gt; In the next few years, La Botz appeared in [[Terry Zwigoff]]'s arthouse film ''[[Ghost World (film)|Ghost World]]''&lt;ref name=ghost-world/&gt; as well as in Buscemi's [[Lonesome Jim]].&lt;ref name=lonesome-jim/&gt; In 2008, La Botz starred in [[Sylvester Stallone]]'s [[Rambo (2008 film)|Rambo]], in which he performed two songs from his album ''Graveyard Jones'', &quot;The Wishing Well&quot; and &quot;Tiny.&quot;&lt;ref name=&quot;films&quot;/&gt; Other notable film appearances include [[Timur Bekmambetov]]'s ''[[Abraham Lincoln: Vampire Hunter (film)|Abraham Lincoln: Vampire Hunter]]'' and [[Walter Salles]]'s film adaptation of [[Jack Kerouac]]'s ''[[On the Road]],'' in which he performs his song &quot;Hard to Love What You Kill.&quot;&lt;ref name=&quot;ontheroad&quot;/&gt;&lt;ref name=&quot;abelincoln&quot;/&gt; La Botz's 2015 film credits include the lead role in The Grace of Jake, an independent film directed by [[Chris Hicky]], and a starring role in [[Isaac Florentine]]'s action movie Close Range.&lt;ref name=&quot;graceofjake&quot;/&gt;&lt;ref name=&quot;closerange&quot;/&gt;

1439 * Jealous of Lola's reaction to [[Sylvester Stallone]] and [[Arnold Schwarzenegger]], Opus attempts to buff up, resulting in injury to himself and others. (p94, 5 strips)

1440 The quote &quot;Loved by Thousands, Hated by Millions&quot; first appears in the 1975 [[Paul Bartel]]'s movie ''[[Death Race 2000]]'' when a television reporter introduces Machine Gun Joe Viterbo. Machine Gun Joe Viterbo was played by [[Sylvester Stallone]].

1441 In 2016, it was announced that Park would be a host on the [[Netflix]] [[reality show]] ''[[Ultimate Beastmaster]]'' along with comedian [[Seo Kyung-seok]].&lt;ref name=&quot;:1&quot;&gt;{{cite web | last=Prudon |first=Laura |date=May 9, 2016 |title=Netflix Orders Global Competition Series 'Ultimate Beastmaster' from Sylvester Stallone, Dave Broome |url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/ |accessdate=October 5, 2016 |work=Variety}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |date=2017-03-07 |title=[XP 인터뷰①] 박경림 &amp;quot;12 시간 에너지 분출, '미친 여자' 갈다고. 하하&amp;quot; |url=https://www.xportsnews.com/article/826176 |access-date=2024-02-17 |website=www.xportsnews.com |language=ko}}&lt;/ref&gt;

1442 | voices = [[Sylvester Stallone]] (John Spartan)&lt;br /&gt;[[Wesley Snipes]] (Simon Phoenix)

1443 [[Sylvester Stallone]] (John Spartan) and [[Wesley Snipes]] (Simon Phoenix) provided custom speech for this game during [[Dubbing (filmmaking)|ADR]] sessions at Warner Brothers Studios in Los Angeles under the direction of Jon Hey. Hey scored the music of the pinball game in part based upon the movie score by [[Elliot Goldenthal]], but including new music.

1444 Sylvester Stallone, Wesley Snipes, and [[Benjamin Bratt]], three of the film's stars, each received a ''Demolition Man'' pinball machine as a gift; a fourth star, [[Sandra Bullock]], did not.&lt;ref&gt;{{cite web |date=June 14, 2018 |author=Hanna Flint |title=Sandra Bullock says she's owed a 'Demolition Man' pinball machine: 'All the men got one' (exclusive) |url=https://www.yahoo.com/entertainment/sandra-bullock-says-shes-owed-demolition-man-pinball-machine-men-got-one-exclusive-142344062.html |website=Yahoo.com }}&lt;/ref&gt;

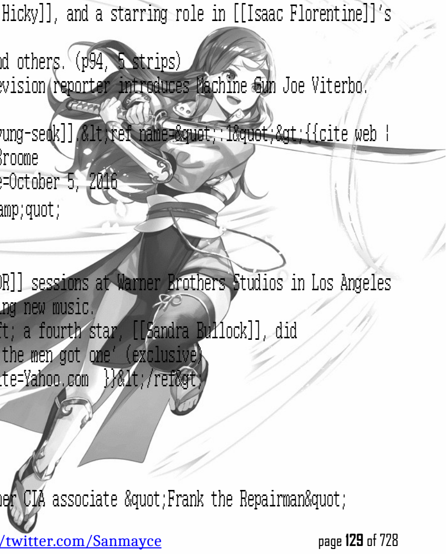
1445 [[Category:Images of Sylvester Stallone]]</text>

1446 * Awarded to [[Sylvester Stallone]] - 1990

1447 [[Category:Images of Sylvester Stallone]]</text>

1448 [[Category:Images of Sylvester Stallone]]</text>

1449 |ShortSummary = When a local restaurateur, and mother of a Montecito showgirl, is approached by petty mobsters, Ed enlists the support of a former CIA associate &quot;Frank the Repairman&quot;



([[Sylvester Stallone]]) to squash their extortion attempts. As the two navigate their way through the local mob's hierarchy, Ed discovers that he has a decades-old connection to the ringleader. Delinda and Nessa rival for the attention of ''Fear Factor'' host Joe Rogan.

* ''[[Demolition Man (film)|Demolition Man]]''&lt;ref&gt;{{cite web|url=https://www.forbes.com/sites/scottmendelson/2015/11/25/the-5-best-sylvester-stallone-movies-that-arent-rocky-or-rambo/#41eeb2649fe9|title=The 5 Best Sylvester Stallone Movies That Aren't 'Rocky' Or 'Rambo'|first=Scott|last=Mendelson|work=Forbes|date=November 25, 2015|access-date=July 1, 2018}}&lt;/ref&gt; (1993)

In [[popular culture]], [[pornography]] (if seen in aesthetic terms) is used as an example of the [[objectification]] of women, and violent films have been cited as the influence for so-called ''[[Copycat crimes|copycat]]'' acts in reality. An example of this was the [[Hungerford Massacre]], where a gunman was alleged by the [[tabloid press]] to have been influenced by the ''[[Rambo (film series)|Rambo]]'' series of films starring [[Sylvester Stallone]].

In 1995, born out of a mutual love for rare and unseen cult films, Bob Murawski and actor/director and son of [[Sylvester Stallone]], [[Sage Stallone]] formed Grindhouse Releasing.&lt;ref name=&quot;SexGore&quot;&gt;[http://www.sexgoremutants.co.uk/spot10.html] Interview with exploitation film distributor Bob Murawski of Grindhouse Releasing&lt;/ref&gt; Murawski continues to run Grindhouse and partner Box Office Spectaculars, both companies that restore, preserve, and distribute classic cult and Euro-horror films. The two labels have digitally remastered classic cult films such as ''[[The Swimmer (1968 film)|The Swimmer]]'' starring [[Burt Lancaster]], ''[[The Big Gundown]]'' starring [[Lee Van Cleef]] and [[Tomas Milian]], Lucio Fulci's spaghetti-horror masterpiece, ''E tu vivrai nel terrore'' (a.k.a. ''[[The Beyond (1981 film)|The Beyond]]'') as well as Italian cannibal films ''Make Them Die Slowly'' (a.k.a. ''[[Cannibal Ferox]]''), ''[[Cannibal Holocaust]]'', American cult film ''[[I Drink Your Blood]]'' (1970) (directed by [[David E. Durston]]),

Continuing to produce and direct documentaries - his 2012 documentary ''Under Fire: Journalists in Combat'' was short-listed for an [[Academy Award]] and won a [[Peabody Award]] in 2012 - his theatrical and cable television film career expanded in Los Angeles and includes the [[Emmy Award]] nominated film for [[TNT (U.S. TV network)|TNT]], ''[[The Pirates of Silicon Valley|Pirates of Silicon Valley]]'' for which he was also nominated for a [[DGA Award]] as director. In 2015 he was awarded the [[International Press Academy]]'s Auteur award. He wrote a number of [[HBO]] and TNT films including ''[[The Second Civil War]]'' starring [[Beau Bridges]] (for which Bridges won an Emmy) and [[James Earl Jones]] and [[Denis Leary]], ''[[Sugartime]]'' starring [[John Turturro]] about [[Chicago mafia]] don [[Sam Giancana]], and an adaptation of [[George Orwell]]'s ''[[Animal Farm]]''. His feature film credits include co-writing the [[Paramount Pictures]] cult classic comedy, ''[[Top Secret!]]'' and directing ''[[Avenging Angelo]]'' starring [[Sylvester Stallone]], [[Anthony Quinn]] and [[Madeline Stowe]].

In the summer of 1985, SCMA cadets were extras in the Sylvester Stallone movie, ''[[Over the Top (1987 film)|Over the Top]]''.&lt;ref&gt;{{cite news |last1=Hale-Burns |first1=Pamela |title=All-class reunion being held for former Signal Hill military academy |url=https://www.presstelegram.com/2010/07/21/all-class-reunion-being-held-for-former-signal-hill-military-academy/ |access-date=26 March 2024 |work=Press-Telegram |archive-url=https://web.archive.org/web/20240326003519/https://www.presstelegram.com/2010/07/21/all-class-reunion-being-held-for-former-signal-hill-military-academy/ |archive-date=26 March 2024}}&lt;/ref&gt;

**[[Sylvester Stallone]]&amp;nbsp;- ''[[Rocky]]'' as [[Rocky Balboa]]

**''[[Rocky]]''&amp;nbsp;- [[Sylvester Stallone]]

* He likes [[Sylvester Stallone]]'s films.

* [[Over the Top (1987 film)|''Over the Top'' (1987 film)]], an action drama film starring Sylvester Stallone

Immigrants from Ukraine were the parents or grandparents of [[Serge Gainsbourg]], [[Leonard Nimoy]], [[Vera Farmiga]], [[Taissa Farmiga]], [[Steven Spielberg]], [[Dustin Hoffman]], [[Sylvester Stallone]], [[Kirk Douglas]], [[Leonardo DiCaprio]], [[Winona Ryder]], [[Whoopi Goldberg]], [[Edward Dmytryk]], [[Lenny Kravitz]] and [[Zoë Kravitz]], illusionist [[David Copperfield (illusionist)|David Copperfield]], animator [[Bill Tytla]].

[[Category:Images of Sylvester Stallone]]</text>

Zayas played a corrupt police officer in ''[[16 Blocks]]'' (2006) and an NYPD officer in ''[[Michael Clayton (film)|Michael Clayton]]'' (2007).{{Citation needed|date=October 2020}} He is featured in the [[independent film]] ''[[Shadowboxing (2010 film)|Shadowboxing]]'' (2010), which appeared on the [[film festival]] circuit. He appears in [[Sylvester Stallone]]'s feature film ''[[The Expendables (2010 film)|The Expendables]]'' (2010) and [[Brothers Strause|The Brothers Strause]]'s film ''[[Skyline (2010 film)|Skyline]]'' (2010). In 2012, Zayas played Ernie Trask, a hotel super, in the twelfth episode of season 1 of ''[[Person of Interest (TV series)|Person of Interest]]''. In the 2013 he portrayed Detective Esteban Flores in ''[[Jodi Arias: Dirty Little Secret]]'', a television movie about the [[murder of Travis Alexander]].&lt;ref&gt;{{Cite web|url=http://thefutoncritic.com/news/2013/05/16/lifetime-original-movie-jodi-arias-dirty-little-secret-starring-tania-raymonde-and-jesse-lee-soffer-to-make-its-world-premiere-saturday-june-22-504514/20130516lifetime01/|title=Breaking News - Lifetime Original Movie &quot;Jodi Arias: Dirty Little Secret,&quot; Starring Tania Raymonde and Jesse Lee Soffer, to Make Its World Premiere Saturday, June 22 &amp;#124; TheFutonCritic.com|website=thefutoncritic.com|access-date=2020-10-11|archive-date=2013-06-16|archive-url=https://archive.today/20130616023959/http://thefutoncritic.com/news/2013/05/16/lifetime-original-movie-jodi-arias-dirty-little-secret-starring-tania-raymonde-and-jesse-lee-soffer-to-make-its-world-premiere-saturday-june-22-504514/20130516lifetime01/|url-status=dead}}&lt;/ref&gt;

Traditionally the domain of cattle ranches, agriculture and small fishing villages, the Pedasí district is now attracting the attention of tourism.&lt;ref&gt;[http://www.elsiglo.com/siglov2/Nacion.php?idnews=64577&amp;idsec=1&amp;fechaz=21-02-2008 &quot;Precarismo impide desarrollo en Azuero (Squatters Hinder Development in Azuero)&quot;], ''[[El Siglo (Panama)|El Siglo]]''. (Spanish) Retrieved Feb 12 2009.&lt;/ref&gt; The area offers extensive [[recreational fishing|sport-fishing]], [[scuba diving]], [[surfing]], and [[birdwatching]]. Pedasí has had celebrity visits by the likes of [[Mick Jagger]],&lt;ref&gt;Machan, T. (2008) [https://web.archive.org/web/20100502004445/http://www.telegraph.co.uk/property/overseasproperty/3394886/Property-in-Panama-Wherever-they-lay-their-hat.html &quot;Property in Panama: Wherever they lay their hat?&quot;], ''Telegraph.co.uk''. Retrieved Feb 12 2009.&lt;/ref&gt; [[Sylvester Stallone]], [[Sean Connery]], [[Mel Gibson]] and [[Tobey Maguire]] and Bryon Beckwith.&lt;ref&gt;Velásquez, A. (2008) [http://www.pa-digital.com.pa/periodico/buscador/resultado.php?story_id=652059 &quot;Not Famous in Pedasí (En Pedasí no son famosos)&quot;], ''Panamá América''. (Spanish) Retrieved Feb 12 2009.&lt;/ref&gt;[[File:Playa Venao from the hills of the Pedasí district, Panama, September 2008.jpg|thumb|upright|220px|A view on Playa Venao in the district of Pedasí, province of Los Santos, Panama, from the surrounding hills.]]

She has interviewed [[Anthony Hopkins]], Benicio del Toro, Sylvester Stallone, [[Salma Hayek]], [[Arnold Schwarzenegger]], [[Vicente Fox]], [[Tony Saca]], former US treasurer [[Rosario Marin]], former [[President of Honduras]] [[Ricardo Maduro]] and his ex-wife [[Aguas Santas Ocaña Navarro]], former Honduran President [[Mel Zelaya|Jose Manuel Zelaya Rosales]] and first Lady Xiomara de Zelaya, [[Alvaro Colom Caballeros]], Guatemala's former President and current Honduras President [[Juan Orlando Hernandez]] among other Latin American Presidents and [[Los Angeles]] former Mayor [[Antonio Villaraigosa]] .

      <text bytes="11873" sha1="hozle723sh5yo4e57ut5i0begbvkwib" xml:space="preserve">{{Short description|Fictional character created by Sylvester Stallone}}

| creator   = [[Sylvester Stallone]]

'''Mickey Goldmill''' is a fictional character created by [[Sylvester Stallone]] and portrayed by [[Burgess Meredith]] in the [[Rocky (franchise)|''Rocky'' film series]]. Meredith was nominated for the [[Academy Award for Best Supporting Actor]] for his performance in the first film, as was his co-star [[Burt Young]]. The character's gravelly voice, intense demeanor and popular catchphrases helped make him highly recognizable as well as a common source of parody and satire in pop culture.&lt;ref&gt;

The 1990s also saw Egyptian cinema participating in the [[martial arts film|martial arts genre]] with actors such as [[Youssef Mansour (actor)|Youssef Mansour]] who became famous for his films that relied on [[martial arts]].&lt;ref&gt;{{cite web | url=https://elcinema.com/en/person/1029757/ | title=Youssef Mansour - Actor Filmography, photos, Video }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=http://www.mafhoum.com/press5/148S63.htm | title=Cairo Times }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://www.masrawy.com/arts/zoom/details/2023/9/30/2474934/بعد-15-عام-ا-على-وفاتها-يوسف-منصور-يفجر-مفاجأة-عن-سوزان-تميم?amp | title=بعد 15 عامًا على وفاتها.. يوسف منصور يفجر مفاجأة عن سوزان تميم }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://cairoscene.com/buzz/stallone-star-in-youssef-mansour-new-movie | title=Sylvester Stallone to Star in Egyptian film with Youssef Mansour }}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=https://scoopempire.com/american-film-chosen-guard-shot-egypt-featuring-youssef-mansour/ | title=American Film 'The Chosen Guard' to be Shot in Egypt Featuring Youssef Mansour | date=12 February 2017 }}&lt;/ref&gt;

Among the special guests who made periodic appearances were [[Miami Dolphins]] football coach [[Don Shula]] and [[Jackie Gleason]]. Every star in town was invited on the show.....[[Sylvester Stallone]], [[Muhammad Ali]], [[Art Linkletter]], several astronauts, [[Barbara Feldon]], [[Carlos Montoya]], [[Van Cliburn]], and many more over the years.

Balboa ([[Sylvester Stallone]]) and Creed ([[Carl Weathers]]) nearly go the distance again in their rematch in [[1979 in film|1979]]'s ''[[Rocky II]]'', as do Rocky and [[Ivan Drago]] in their showdown in ''[[Rocky IV]]''. Balboa's final fight against [[Mason Dixon (Rocky Balboa character)|Mason &quot;The Line&quot; Dixon]] in ''[[Rocky Balboa (film)|Rocky Balboa]]'' lasts the maximum of 10 rounds.

*In ''[[Command &amp; Conquer: Yuri's Revenge]]'', Sylvester Stallone (voice [[Impersonator|impersonated]]) is featured as an in-game celebrity character, &quot;Sammy Stallion&quot;, who frequently says, &quot;I'm goin' the distance&quot;, when directed to move in the battlefield of the Hollywood mission.&lt;ref&gt;''[[Command &amp; Conquer: Yuri's Revenge]]'' (2001) [[Westwood Studios]]. [[EA Games]]&lt;/ref&gt;&lt;ref&gt;[https://www.imdb.com/title/tt0075148/movieconnections Movie connections for Rocky (1976)]. IMDB&lt;/ref&gt;&lt;ref&gt;Westlake &quot;[http://www.game-over.net/reviews.php?id=665 Game Reviews: Command &amp; Conquer: Yuri's Revenge].&quot; Moby Games.&lt;/ref&gt;

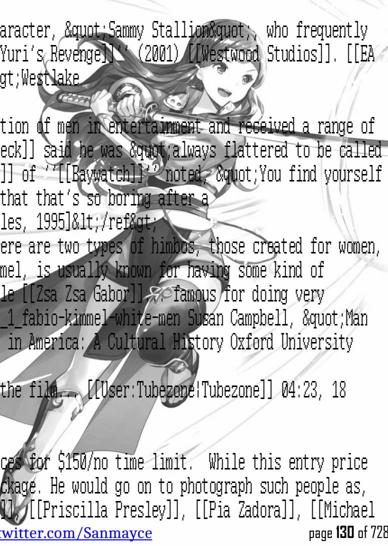
In 1995, [[Sherri Sylvester]] of [[CNN]] interviewed male Hollywood celebrities about the use of the term &quot;himbo&quot; and sexual objectification of men in entertainment and received a range of reactions. &quot;There's a great word,&quot; said actor [[Keanu Reeves]]. &quot;I love that. I read that and laughed my head off.&quot; [[Tom Selleck]] said he was &quot;always flattered to be called a sex symbol&quot; but [[Sylvester Stallone]] said he had fought &quot;the stereotype that brawny means brainless&quot; for years. [[David Charvet]] of ''[[Baywatch]]'' noted, &quot;You find yourself doing a show for three years where you are sticking your chest out and your shoulders are back and you're holding in your stomach and you realize that that's so boring after a while.&quot;&lt;ref&gt;Sherri Sylvester, &quot;[http://www.cnn.com/SHOWBIZ/Movies/hunks Hollywood Hunks-Just a Bunch of Himbos&quot;, CNN Los Angeles, 1995]&lt;/ref&gt;

In a 1994 interview, sociologist Michael Kimmel, who analyzes the himbo stereotype in his book ''Manhood in America: A Cultural History'', said there are two types of himbos, those created for women, like the model [[Fabio Lanzoni|Fabio]], and those created for men, like [[Arnold Schwarzenegger]] or Sylvester Stallone. The man's himbo, says Kimmel, is usually known for having some kind of prowess, like [[Charles Atlas]] or Stallone, whereas the woman's himbo is 'kinder and gentler' like [[Woody Harrelson]] or, like Fabio, &quot;a male [[Zsa Zsa Gabor]] - famous for doing very little.&quot;&lt;ref name=Sentinel&gt;[https://web.archive.org/web/20170202012237/http://articles.sun-sentinel.com/1994-06-16/lifestyle/9406140552_1_fabio-kimmel-white-men Susan Campbell, &quot;Man As Object Becomes A Himbo&quot;, ''The Sun-Sentinel'', June 16, 1994], retrieved January 23, 2017&lt;/ref&gt;&lt;ref&gt;Michael S. Kimmel, Manhood in America: A Cultural History Oxford University Press, 2012 -&lt;/ref&gt;

:Adding [[Gay For A Summer]] to this AfD, these should be deleted together, unless [[Sylvester Stallone]] suddenly pops up to confirm his role in the film... [[User:Tubezone|Tubezone]] 04:23, 18 December 2006 (UTC)

* ''[[Avenging Angelo]]'' (2002); Sylvester Stallone

Fong, left with no mentor or proper training, started a wedding photography business in a bedroom of his parents' apartment, and offered his services for $150/no time limit. While this entry price severely undermined the pricing standards, he was able to present comprehensive designs that were far more costly than the standard photography package. He would go on to photograph such people as, [[Sylvester Stallone]], [[Ronald Reagan]], [[Sidney Sheldon]], [[Lee Remick]], [[Paul McCartney]], [[Peter Criss]], [[Bruce Lee]], [[Todd Rundgren]], [[Priscilla Presley]], [[Pia Zadora]], [[Michael



Landon]], [[Florence Henderson]], [[Shelly Fabares]], among others.&lt;ref&gt;{{Cite web |last=POHREN |first=APRIL |date=2010-07-12 |title=Interview: Gary Fong, Author of The Accidental Millionaire:... |url=https://www.seattlepi.com/lifestyle/blogcritics/article/Interview-Gary-Fong-Author-of-The-Accidental-886665.php |access-date=2022-12-12 |website=Seattle Post-Intelligencer | language=en}}&lt;/ref&gt;

1476 * ''[[Rambo (2008 film)|Rambo]]'' (2008) - [[John Rambo]], the main protagonist of the series, played by [[Sylvester Stallone]] joins a team of mercenaries, on a mission to save a group of volunteered missionaries, who are held captive in Burma.

1477 * ''[[The Expendables (franchise)|The Expendables]]'' series of movies about a group of elite mercenaries, starring [[Sylvester Stallone]], [[Jason Statham]], [[Dolph Lundgren]], [[Jet Li]], [[Mickey Rourke]], [[Randy Couture]], [[Stone Cold Steve Austin|Steve Austin]], [[Terry Crews]], [[Bruce Willis]], [[Arnold Schwarzenegger]], [[Jean-Claude Van Damme]], [[Liam Hemsworth]], [[Chuck Norris]], [[Antonio Banderas]], [[Wesley Snipes]], [[Mel Gibson]], [[Harrison Ford]], [[Kelsey Grammer]], [[Ronda Rousey]], [[Tony Jaa]], and [[50 Cent]].

1478 He later acted in the TV play ''Vita col padre e con la madre'' with [[Paolo Stoppa]], [[Rina Morelli]] and [[Corrado Pani]]. During that time, Sorrentino started his career as a voice actor, dubbing a role in Italian in [[Delmer Daves]]' 1956 film ''[[The Last Wagon (1956 film)|The Last Wagon]]''.&lt;ref&gt;{{cite web|url=https://www.mymovies.it/biografia/?a=4462 |title=Claudio Sorrentino | author= |publisher=MYmovies |date= |access-date= 26 March 2020}}&lt;/ref&gt; As a dubbing artist, Sorrentino was the official Italian voice of [[John Travolta]] and [[Mel Gibson]].&lt;ref&gt;[https://www.youtube.com/watch?v=mhyzRracA3I Intervista a CLAUDIO SORRENTINO (2015) | enciclopediadeldoppiaggio.it]&lt;/ref&gt; Other actors he dubbed include [[Bruce Willis]], [[Sylvester Stallone]], [[Willem Dafoe]], [[Ryan O'Neal]], [[Ron Howard]], [[Jeff Bridges]], [[Mickey Rourke]] and many more. In his animated roles, he voiced Dodger in the Italian dub of ''[[Oliver &amp; Company]]'', ''[[Gargamel]]'' in ''[[The Smurfs (1981 TV series)|The Smurfs]]'', and [[Mickey Mouse]] from 1978 to 1981.

1479 |[[Sylvester Stallone]]

1480 In 1989, Jamison contributed his own version of &quot;[[Ever Since the World Began]]&quot;, a song Survivor had initially recorded prior to his tenure in the band, to the Sylvester Stallone's film, ''[[Lock Up (1989 film)|Lock Up]],'' after being requested by Stallone. That same year, he was invited to join [[Deep Purple]], who had just fired longtime singer [[Ian Gillan]]. After a two-week session with the band, he reluctantly parted ways with them; according to Deep Purple organist [[Jon Lord]], &quot;He [Jamison] was an enormous Deep Purple fan and he would happily have taken over the job. But at the time he was afraid of his managers. They didn't want him to leave [Survivor] and he didn't dare to get into a fight with them.&quot; In fact, Jamison's managers wanted him to concentrate on his upcoming solo album instead of joining Deep Purple.&lt;ref&gt;{{cite web |url=http://www.picturedwithin.com/interviews/tbro_int.html |title=Jon Lord Interview at | publisher=Picturedwithin.com |date=October 4, 1993 |access-date=May 13, 2013 |archive-url=https://web.archive.org/web/20100517061304/http://www.picturedwithin.com/interviews/tbro_int.html |archive-date=May 17, 2010 |url-status=dead }}&lt;/ref&gt; Also in 1989, Jamison briefly united with his Target bandmates for a &quot;10-Year Reunion&quot; concert in Memphis and was part of the New Year's Eve celebration in the Crown Plaza. It was also around this time that Jamison performed the entrance song &quot;Hard Time&quot; for the wrestler [[Big Boss Man (wrestler)|Big Boss Man]] when he was in the [[World Wrestling Federation|WWF]].&lt;ref&gt;{{Cite web|title=MR • View topic - Jimi Jamison When The Phone Don't Ring???|url=http://forums.melodicrock.com/phpBB/viewtopic.php?t=18417|access-date=September 6, 2018|website=forums.melodicrock.com|language=en-gb}}&lt;/ref&gt;

1481 |[[Sylvester Stallone]] ||''[[Creed (film)|Creed]]''||[[Rocky Balboa]]

1482 Part of the Sylvester Stallone film ''[[The Expendables (2010 film)|The Expendables]]'' was shot in the central part of the city. The pier that is exploded in the film is the pier that the main ferry to [[Ilha Grande]] uses as its base of operation.

1483 Woodcock has interviewed many A-list actors and bold-faced names including Barbara Walters, Ben Affleck, Orlando Bloom, Leonardo DiCaprio, Donald Trump, Keith Olbermann, Jack Valenti, Paris, Kathy, and Nikki Hilton, Rosie O'Donnell, Arianna Huffington, Patrick Dempsey, Ellen Barkin, Matthew Perry, Hilary Swank, Eva Longoria, John Voight, James Caan, Brooke Shields, [[Forest Whitaker]], Vanna White, Miley Cyrus, Ellen Pompeo, George Clooney, Brad Pitt, Matt Damon, Don Cheadle, Rosanna Arquette, Danni and Anjelica Huston, Hugh Hefner, Bill Clinton, Al Gore, Marcia Cross, Denise Richards, Tom Hanks, Sylvester Stallone, Bruce Willis, Eric Dane, Aaron Eckhart, Billy Bob Thornton, Jeremy Piven, Bill Paxton, Jeremy Irons, Mark Whalberg, Heidi Klum, Clive Davis, Jennifer Hudson, Keiffer Sutherland, Minnie Drive, Jolie Fisher, Kyra Sedwick, Alejandro Gonzales Innarutu, Matt Dillon, Harry Connick Jr., Hilary Swank, Jeffry Dean Morgan, Roger King, and Gerard Butler.{{citation needed| date=October 2016}}

1484 :::::For [[User:Caglarkoca|Caglarkoca]], the links I gave were to show how much of an impact the Armenian Genocide has on Turkish foreign relations. It affected French relations, it has affected relations with every country that has accepted the events of 1915 as genocide. I am not asking for an article on US, France etc. I am asking something simple that is very clear. The impact of the Armenian Genocide on Turkey is big. There are demonstrations by nationalists in Turkey against it, the film Ararat was banned in Turkey, the upcoming ban on Sylvester Stallone's 40 days of Musa Dagh film, Orhan Pamuk's arrest, the cancelled military contracts w/ France, the closed border w/ Armenia, some EU countries questioning of Turkey's place in the EU because of the Armenian Genocide...these are all the because of the Armenian Genocide issue. I think it deserves a simple mention. I am confused on why this is not encyclopedic. Besides on a side note, do you think Turkey would be pleased if the Democrats pass a bill recognizing the genocide? Anyway this is off topic, I have said my points. Thanks [[User:Fedayee|Fedayee]] 23:54, 29 December 2006 (UTC)

1485 *[[Sylvester Stallone]] as [[Rocky Balboa]]

1486 McTavish's next feature film was 2002's ''[[Ali G Indahouse]],'' where he portrayed a customs officer, followed by 2003's ''[[Dot the i]]'', where he portrayed a detective opposite [[Tom Hardy]].&lt;ref&gt;{{cite web|title=Ali G Indahouse (2002)|url=https://www.bfi.org.uk/films-tv-people/4ce2b8673a57e|archive-url=https://web.archive.org/web/20190503014053/http://www.bfi.org.uk/films-tv-people/4ce2b8673a57e|url-status=dead|archive-date=3 May 2019|access-date=9 April 2019|website=BFI|language=en}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=Dot The I {{!}} TV Guide|url=https://www.tvguide.com/movies/dot-the-i/cast/137865|access-date=9 April 2019|website=TVGuide.com|language=en}}&lt;/ref&gt; He went on to portray a submarine captain in director [[Jan de Bont]]'s ''[[Lara Croft: Tomb Raider - The Cradle of Life|Lara CroftTomb Raider: The Cradle of Life]]''.&lt;ref&gt;{{cite web|title=Lara Croft Tomb Raider: The Cradle of Life (2003) - Overview - TCM.com|url=http://www.tcm.com/tcmdb/title/448180/Lara-Croft-Tomb-Raider-The-Cradle-of-Life/|access-date=11 April 2019|website=Turner Classic Movies|language=en}}&lt;/ref&gt; The next year he was featured in [[Walt Disney Studios Motion Pictures|Buena Vista Pictures]]' action adventure film ''[[King Arthur (2004 film)|King Arthur]]''.&lt;ref&gt;{{Citation|title=King Arthur (2004)| url=https://www.rottentomatoes.com/m/1133964_king_arthur|language=en|access-date=17 April 2019}}&lt;/ref&gt; In 2008, McTavish starred opposite [[Sylvester Stallone]] in ''[[Rambo (2008 film)| Rambo]]'', the fourth film in the franchise based upon [[David Morrell]]'s 1972 novel [[First Blood (novel)|''First Blood'']].&lt;ref&gt;{{cite web|last=Metz|first=Nina|title=My worst moment: Graham McTavish and a bovine co-star|url=https://www.chicagotribune.com/entertainment/tv/ct-ent-my-worst-moment-graham-mctavish-0725-story.html|access-date=17 April 2019|website=Chicago Tribune|date=24 July 2018 |language=en-US}}&lt;/ref&gt; His next film, director Richard Wellings-Thomas' comedy ''[[Sisterhood (2008 film)|Sisterhood]]'', saw McTavish in the role of Martin.&lt;ref&gt;{{cite web| title=Graham McTavish Wizard World Comic Con|url=https://wizardworld.com/index.php/guests/graham-mctavish|access-date=18 April 2019|website=Wizard World Comic Con|language=en}}&lt;/ref&gt; 2009 saw several film roles for McTavish, including prison drama ''[[Green Street 2: Stand Your Ground|Green Street 2]],'' independent horror film ''[[Penance (2009 film)|Penance]],'' Jason Connery's thriller ''Pandemic,'' and the dramatic comedy ''[[Middle Men (film)|Middle Men]]'', opposite [[Luke Wilson]].&lt;ref&gt;{{cite web|title=Green Street 2: Stand Your Ground - review {{!}} cast and crew, movie star rating and where to watch film on TV and online|url=http://www.radiotimes.com/film/t82v/green-street-2-stand-your-ground/|access-date=30 January 2018|website=Radio Times|language=en}}&lt;/ref&gt;&lt;ref&gt;{{Citation|title=Green Street Hooligans 2: Stand Your Ground (2009)|url=https://www.rottentomatoes.com/m/1213418_green_street_hooligans_2|language=en|access-date=18 April 2019}}&lt;/ref&gt;&lt;ref&gt;{{Citation|title=Penance (2009)|url=https://www.rottentomatoes.com/m/penance|language=en|access-date=18 April 2019}}&lt;/ref&gt;&lt;ref&gt;{{Citation| title=Pandemic (2008)|url=https://www.rottentomatoes.com/m/pandemic_2009|language=en|access-date=18 April 2019}}&lt;/ref&gt;&lt;ref&gt;{{Citation|title=Middle Men (2010)|url=https://www.rottentomatoes.com/m/middle_men|language=en|access-date=18 April 2019}}&lt;/ref&gt;

1487 Between projects, Irvin has served as an adjunct faculty member at the University of Southern California School of Cinematic Arts where he taught graduate courses on directing. Among his former students is [[Ryan Coogler]], acclaimed writer-director of ''[[Fruitvale Station (film)|Fruitvale Station]]'', winner of the Grand Jury Prize and Audience Award at the 2013 Sundance Film Festival, as well as the Un Certain Regard award at the 2013 Cannes Film Festival. Coogler has since directed the acclaimed hit ''[[Creed (film)|Creed]]'' starring [[Michael B. Jordan]] and [[Sylvester Stallone]]; and the [[Marvel superhero]] movie ''[[Black Panther (film)|Black Panther]]''.

1488 [[Category:Images of Sylvester Stallone]]</text>

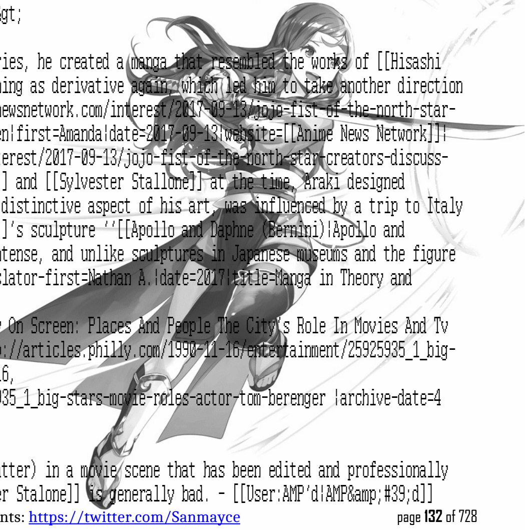
1489 [[Category:Images of Sylvester Stallone]]</text>

1490 Tarantino attempted to cast [[John Travolta]], [[Denzel Washington]], [[Willem Dafoe]], [[John Malkovich]], [[Mickey Rourke]], [[Ron Perlman]], [[Bruce Willis]], [[Kal Penn]],&lt;ref&gt;{{cite web| url=http://www.slashfilm.com/article.php?story=20061121kalpenngrindhouse |archive-url=https://archive.today/20070722151341/http://www.slashfilm.com/article.php?story=20061121kalpenngrindhouse | archive-date=July 22, 2007 |title=Did You Know: Kal Penn was cast in Tarantino's ''Grindhouse''? |date=November 21, 2006 |access-date=January 6, 2007 |first=Peter |last=Sciretta |website=[[/Film]]}}&lt;/ref&gt; and [[Sylvester Stallone]]&lt;ref&gt;{{cite web|url=http://www.slashfilm.com/article.php?story=20061204stallonegrindhouse |title=Tarantino wanted Stallone for ''Grindhouse'' |date=December 4, 2006 |access-date=January 6, 2007 |first=Peter |last=Sciretta |website=/Film |archive-url=https://web.archive.org/web/20110511145304/http://www.slashfilm.com/ article.php?story=20061204stallonegrindhouse |archive-date=May 11, 2011}}&lt;/ref&gt; in ''Death Proof'', but none were able to work due to prior commitments. In an interview, Tarantino revealed that he cast [[Kurt Russell]] as the killer stunt driver because &quot;for people of my generation, he's a true hero ... but now, there's a whole audience out there that doesn't know what Kurt Russell can do. When I open the newspaper and see an ad that says 'Kurt Russell in ''[[Dreamer (2005 film)|Dreamer]]'',' or 'Kurt Russell in ''[[Miracle (2004 film)|Miracle]]'',' I'm not disparaging these movies, but I'm thinking: When is Kurt Russell going to be a badass again?&quot;&lt;ref name=&quot;Bloodbath&quot;&gt;{{Cite magazine | last =Nashawaty | first =Chris | title =Bloodbath and Beyond | newspaper =[[Entertainment Weekly]] | pages =27-30 | date =March 30, 2007 | url =https://ew.com/article/2007/03/27/hungry-zombies-psychopath-killer-car-its-grindhouse/ | access-date =September 25, 2007 | archive-date =July 26, 2014 | url-status =live | archive-url =https://web.archive.org/web/20140726123034/http://www.ew.com/ew/article/0,,20015706,00.html}}&lt;/ref&gt; [[Eli Roth]], ''Planet Terror'' leading actress Rose McGowan, and Tarantino himself appear in the film. Roth flew in from Europe, where he was filming ''[[Hostel: Part II]]'', to film his scenes, which took one day.

1491 | caption = North American SG cover artwork featuring [[Sylvester Stallone]] as [[Judge Dredd]]

1492 Cuevas's client list includes [[Hank Williams]], [[Waylon Jennings]], [[Porter Wagoner]], [[John Wayne]], [[Clayton Moore]] ([[the Lone Ranger]]), [[Dwight Eisenhower]], [[Little Jimmy Dickens]], [[John Lennon]], [[Loretta Lynn]], [[George Jones]], [[Glen Campbell]], [[Ernest Tubb]], [[Gene Autry]], [[the Osmonds]], [[David Cassidy]], [[Bobby Sherman]], [[Dolly Parton]], [[Linda Ronstadt]], [[Emmylou Harris]], [[Roy Rogers]], [[Neil Young]], [[Elton John]], [[The Grateful Dead]], [[The Rolling Stones]], [[Bob Dylan]], [[George H. Bush]], [[George W. Bush]], [[the Bee Gees]], [[Janis Joplin]], [[Jimi Hendrix]], [[Catherine Bach]] ([[Daisy Duke]]), [[The Jackson Five]], [[John Travolta]] ([[Urban Cowboy]]), [[Robert Redford]] (''[[The Electric Horseman]]''), [[Robert Taylor (American actor)|Robert Taylor]], [[Marlon Brando]], [[Burt Reynolds]], [[Raquel Welch]], [[David Lee Roth]], Clint Eastwood in Sergio Leone's Dollars Trilogy of Spaghetti Westerns (''A Fistful of Dollars'', ''For a Few Dollars More'' and ''The Good, the Bad and the Ugly''), [[Jack Nicholson]], [[Sylvester Stallone]], [[Mike Mills]], [[Shooter Jennings]], [[Kid Rock]], [[The Killers]], [[Jack White]], [[Kenny Chesney]], [[Randy Travis]], [[Alan Jackson]], [[Tim McGraw]], [[Keith Urban]], [[Zac Brown Band]], [[Miranda Lambert]], [[Jon Pardi]], [[Frankie Ballard]], [[Ozzy Osbourne]], [[Dalton Grap]], and countless others.&lt;ref name=Lo/&gt;&lt;ref name=&quot;Jackets Required&quot;/&gt;&lt;ref name=Hollabaugh/&gt;&lt;ref name=Quintanilla/&gt;&lt;ref name=Schulman&gt;{{cite web| last1=Schulman|first1=Sandra|title=How the West is Worn|url=http://articles.sun-sentinel.com/1993-08-15/features/9301300038_1_manuel-cuevas-jackets-new-image|website=SunSentinel.com|publisher=Sun Sentinel|access-date=7 October 2014|archive-date=11 October 2014|archive-url=https://web.archive.org/web/20141011212348/http://articles.sun-sentinel.com/1993-08-15/features/9301300038_1_manuel-

```
cuevas-jackets-new-image|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last1=Oatman-Stanford|first1=Hunter|title=Meet the Man Who Made Cowboys Love Rhinestones|url=http://www.collectorsweekly.com/articles/meet-the-man-who-made-cowboys-love-rhinestones/|website=Collectors Weekly.com|publisher=Collectors Weekly}}&lt;/ref&gt;&lt;ref name=&quot;Punk Globe&quot;&gt;{{cite web|last1=Smith|first1=Kevin|title=The Rhinestone Rembrandt: Manuel|url=http://punkglobe.com/manuelinterview0814.php|website=PunkGlobe.com|publisher=Punk Globe}}&lt;/ref&gt;
*November 21, 1976 - The film '''''[[Rocky]]''''', starring [[Sylvester Stallone]], is released in U.S. theaters.
*The [[Woody Allen]] film ''[[Bananas (film)|Bananas]]'' (also 1971) has a scene in which Woody's character Fielding Melish confronts two muggers (one played by a pre-fame [[Sylvester Stallone]]) assaulting an old lady by tossing them off the train. As with the above example, this scene was shot on [[42nd Street Shuttle]] at Grand Central.
| 669 ||September 27 ||[[Sylvester Stallone]] || ''[[Get Carter (2000 film)|Get Carter]]''
&quot;'''Burning Heart'''&quot; is a song by [[Survivor (band)|Survivor]]. It appeared in the [[1985 in film|1985]] film ''[[Rocky IV]]'' and on its [[Rocky IV (soundtrack)|soundtrack album]]; the film's star [[Sylvester Stallone]] personally commissioned the song.&lt;ref&gt;{{cite book |first=Alex P. |last=King |title=Hit-parade - 20 ans de tubes |publisher=Pascal |year=2004 |place=Paris |page=342 |language=fr |isbn=2-35019-009-9}}&lt;/ref&gt; The [[Single (music)|single]] peaked at number 2 on the [[Billboard Hot 100|''Billboard'' Hot 100]] for two weeks in February 1986, behind &quot;[[That's What Friends Are For]]&quot; by [[Dionne Warwick|Dionne]] and Friends. It was the biggest hit the band had with [[Jimi Jamison]] on lead vocals.  It was also a top 5 hit in the UK, Germany and several other European countries, and reached the top 15 in Canada.
    <title>Salvestor Stallone</title>
    <redirect title="Sylvester Stallone" />
      <comment>[[WP:AES|←]]Redirected page to [[Sylvester Stallone]]</comment>
      <text bytes="32" sha1="ie7gor6h9px5f78sr4gwf1p3kp8dcmd" xml:space="preserve">#Redirect [[Sylvester Stallone]]</text>
The nominations were announced on January 22, 2008, at the [[Samuel Goldwyn Theater]] in [[Beverly Hills, California]] by [[Sid Ganis]], president of the Academy, and actress [[Kathy Bates]].&lt;ref&gt;{{Cite news |last=Gray |first=Tim |date=January 22, 2008 |title=Oscar nominations announced |work=Variety |url=https://variety.com/2008/film/news/oscar-nominations-announced-1117979394/ |url-status=live |access-date=June 26, 2014 |archive-url=https://web.archive.org/web/20151223094924/http://variety.com/2008/film/news/oscar-nominations-announced-1117979394/ |archive-date=December 23, 2015}}&lt;/ref&gt; ''[[No Country for Old Men]]'' and ''[[There Will Be Blood]]'' tied for the most nominations with eight each.&lt;ref&gt;{{Cite news |last=Barnes |first=Brooks |date=January 23, 2008 |title='No Country' and 'There Will Be Blood' Lead Oscars |work=The New York Times |url=https://www.nytimes.com/2008/01/23/movies/awardsseason/23oscarscnd.html |url-status=live |access-date=June 26, 2014 |archive-url=https://web.archive.org/web/20160717122608/http://www.nytimes.com/2008/01/23/movies/awardsseason/23oscarscnd.html |archive-date=July 17, 2016}}&lt;/ref&gt; [[Cate Blanchett]] became the eleventh performer to receive [[List of actors nominated for two Academy Awards in the same year|two acting nominations in the same year]],&lt;ref name=&quot;EW&quot;&gt;{{Cite magazine |last=Karger |first=Dave |date=January 25, 2008 |title=The unpredictable 2008 Oscars |magazine=[[Entertainment Weekly]] |url=https://www.ew.com/article/2008/01/25/unpredictable-2008-oscars |url-status=live |access-date=February 3, 2008 |archive-url=https://web.archive.org/web/20151122194115/http://www.ew.com/article/2008/01/25/unpredictable-2008-oscars |archive-date=November 22, 2015}}&lt;/ref&gt; as well as being the first actress and fifth performer overall to be nominated for portraying the same character in two different films, by virtue of her nomination for her role as [[Elizabeth I of England]] in ''[[Elizabeth: The Golden Age]]'' (she had previously been nominated for playing Elizabeth I in 1998's ''[[Elizabeth (film)|Elizabeth]]'').&lt;ref&gt;[https://www.hollywoodreporter.com/gallery/oscar-nominations-playing-same-character-867131/1-sylvester-stallone Oscars: 6 Actors Nominated for Playing the Same Character Twice] {{Webarchive|url=https://web.archive.org/web/20180714221900/https://www.hollywoodreporter.com/gallery/oscar-nominations-playing-same-character-867131/1-sylvester-stallone |date=July 14, 2018 }} Other actors nominated for the same character in multiple films were&lt;/p&gt;&lt;p&gt;[[Paul Newman]] for &quot;Fast Eddie&quot; Felson in [[The Hustler]] and [[The Color of Money]];&lt;/p&gt;&lt;p&gt;[[Al Pacino]] for [[Michael Corleone]] in [[The Godfather]] and [[The Godfather Part II]];&lt;/p&gt;&lt;p&gt;[[Peter O'Toole]] for King Henry II in [[Becket (1964 film)|Becket]] and [[The Lion in Winter (1968 film)|The Lion in Winter]];&lt;/p&gt;&lt;p&gt;[[Bing Crosby]] for Father O'Malley in [[Going My Way]] and [[The Bells of St. Mary's (1959 film)|The Bells of St. Mary's]];&lt;/p&gt;&lt;p&gt;and later, [[Sylvester Stallone]] for [[Rocky (film)|Rocky]] and [[Creed (film)|Creed]]&lt;/p&gt;&lt;/ref&gt; At age 82, Best Supporting Actor nominee [[Hal Holbrook]] became the oldest male acting nominee in Oscar history at the time.&lt;ref&gt;{{Cite news |date=January 22, 2008 |title=Academy Award Nominee Hal Holbrook |publisher=[[NBC News]] ([[NBCUniversal]]) |url=http://www.today.com/allday/academy-award-nominee-hal-holbrook-1C9385251 |url-status=live |access-date=June 27, 2014 |archive-url=https://web.archive.org/web/20160128101749/http://www.today.com/allday/academy-award-nominee-hal-holbrook-1C9385251 |archive-date=January 28, 2016}}&lt;/ref&gt;
Gombo, a Mongolian shepherd, lives in a [[yurt]] in Inner Mongolia with his wife, three children, and mother. Gombo desires sexual intercourse with his wife, which puts his wife ill at ease: having a fourth child would break Chinese law. Meanwhile, Sergei, a buffoonish Russian truck driver falling asleep at the wheel, stops after a close call and attempts to awaken himself. His stretching turns to frolicking, and he wanders off. He discovers a corpse and quickly returns to his truck, accidentally driving into the river in his haste. Back at the yurt, the family is portrayed as unsophisticated and traditional people. They work together performing simple tasks like weaving by hand and using a horse cart to work leather. Gombo's drunk, horse-riding relative rides through the scene, stopping along the way to give them a movie poster for Cobra, which displays a muscular, gun-toting Sylvester Stallone.
'''David Morrell''' (born April 24, 1943) is a Canadian-American [[author]] whose debut 1972 novel ''[[First Blood (novel)|First Blood]]'', later adapted as the 1982 [[First Blood|film of the same name]], went on to spawn the successful [[Rambo (franchise)|''Rambo'' franchise]] starring [[Sylvester Stallone]].&lt;ref name=&quot;cr&quot; /&gt; He has written 28 novels, and his work has been translated into 30 languages.&lt;ref name=BioPage&gt;{{cite web |url=https://davidmorrell.net/bio/ |title=Biography }}&lt;/ref&gt; He  also wrote the 2007-2008 ''[[Captain America]]'' comic book miniseries ''The Chosen''.
During his time at Penn State he met [[science fiction]] writer Philip Klass, better known by the pseudonym [[William Tenn]], who taught the basics of writing fiction.&lt;ref name=BioPage/&gt; Morrell began work as an English professor at the [[University of Iowa]] in 1970. In 1972, his novel ''[[First Blood (novel)|First Blood]]'' was published; it would eventually be made into the [[First Blood|1982 film of the same name]] starring [[Sylvester Stallone]] as [[Vietnam veteran]] [[John Rambo]].&lt;ref name=&quot;cr&quot; /&gt; Morrell continued to write many other novels, including ''[[The Brotherhood of the Rose]]'', the first in a trilogy of novels, which was adapted into a [[Brotherhood of the Rose (miniseries)|1989 NBC miniseries]] starring [[Robert Mitchum]]. He gave up his tenure at the university in 1986 in order to write full-time.&lt;ref name=BioPage/&gt; In 1988 he received the Horror Writers Association award for best novella; ''Orange Is for Anguish, Blue for Insanity''.&lt;ref&gt;[https://books.google.com/books?id=y4MZBAAAQBAJ&amp;dq=%22Horror+Writers+Association+award%22+best+novella+of+1988&amp;pg=PT237 Black Evening: Tales of Dark Suspense]&lt;/ref&gt;
[[Category:Images of Sylvester Stallone]]</text>
| rowspan=3 | 2017 || Sophia Stallone || rowspan=3 | [[Sylvester Stallone]] and [[Jennifer Flavin]] || 20
Darbois is a regular French voice for many actors, including [[Harrison Ford]], [[Danny Glover]], [[Richard Gere]], [[Jeff Goldblum]], [[Dan Aykroyd]], and [[Patrick Swayze]], as well as for [[Bill Murray]], [[Arnold Schwarzenegger]] and [[Sylvester Stallone]].
Darbois has since been in many French versions for films and television series through animation. He doubles [[Harrison Ford]], [[Richard Gere]], [[Danny Glover]] and [[Jeff Goldblum]] and has repeatedly dubbed [[Bill Murray]], [[Patrick Swayze]], [[Arnold Schwarzenegger]], [[Liam Neeson]], [[George Clooney]] and even [[Sylvester Stallone]] (during a certain period, Alain Dorval, the usual voice of Stallone, was at odds with the studios, following union demands).&lt;ref&gt;{{cite news |title=Haute sécurité : pourquoi la voix française de Sylvester... - Télé Star |url=https://www.telestar.fr/culture/haute-securite-pourquoi-la-voix-francaise-de-sylvester-stallone-est-differente-dans-ce-film-545804 |access-date=8 November 2021 |work=www.telestar.fr |date=24 October 2020 |language=fr}}&lt;/ref&gt;&lt;ref&gt;{{cite news |title=Le portrait de la semaine : la double vie d'Alain Dorval, voix de Sylvester Stallone et chef d'entreprise à Chambly |url=https://france3-regions.francetvinfo.fr/hauts-de-france/oise/le-portrait-de-la-semaine-la-double-vie-d-alain-dorval-voix-de-sylvester-stallone-et-chef-d-entreprise-a-chambly-1954579.html |access-date=8 November 2021 |work=France 3 Hauts-de-France |language=fr-FR}}&lt;/ref&gt;
Jack Nicholson earned a [[Golden Raspberry Award]] nomination for Worst Actor for his performances in both this film and ''[[Hoffa (film)|Hoffa]]'', but lost the trophy to [[Sylvester Stallone]] for ''[[Stop! Or My Mom Will Shoot]]''.&lt;ref&gt;{{cite news |last1=MacMinn |first1=Aleene |title=Movies |url=https://www.latimes.com/archives/la-xpm-1993-02-16-ca-29-story.html |access-date=19 May 2023 |work=[[Los Angeles Times]] |date=1993-02-16}}&lt;/ref&gt;
''The Illusionist'' opened in the United Kingdom in 42 cinemas (August 2010). It entered the UK box office at #15, with revenue of £161,900 one place behind Disney's ''[[Tinker Bell and the Great Fairy Rescue]]'', the chart dominated by [[Sylvester Stallone]]'s ''[[The Expendables (2010 film)|The Expendables]]'' which grossed £3,910,596 in revenue in its first weekend of release.&lt;ref&gt;[https://web.archive.org/web/20160119145616/http://www.ukfilmcouncil.org.uk/article/16959/UK UK film Council], 22 August 2010&lt;/ref&gt;
| Chuck Dixon, Richard Meyer, Sylvester Stallone
''Phantom Blood'' was written and drawn by [[Hirohiko Araki]].&lt;ref name=&quot;ann review pb1&quot;/&gt; Prior to working on the series, he created a manga that resembled the works of [[Hisashi Eguchi]], an artist known for his art of female characters; Araki's editor, Ryōsuke Kabashima, angrily told Araki to never draw something as derivative again, which led him to take another direction with his art, and the creation of ''Phantom Blood''.&lt;ref name=&quot;north star influences&quot;&gt;{{cite web|url=http://www.animenewsnetwork.com/interest/2017-09-13/jojo-fist-of-the-north-star-creators-discuss-their-manga-influences/.121287|title=JoJo, Fist of the North Star Creators Discuss Their Manga, Influences|last=Whalen|first=Amanda|date=2017-09-13|website=[[Anime News Network]]|publisher=2017-09-13|access-date=2017-09-30|archive-url=https://web.archive.org/web/20170914123847/http://www.animenewsnetwork.com/interest/2017-09-13/jojo-fist-of-the-north-star-creators-discuss-their-manga-influences/.121287|archive-date=2017-09-14|url-status=live}}&lt;/ref&gt; Due to the popularity of [[Arnold Schwarzenegger]] and [[Sylvester Stallone]] at the time, Araki designed characters with muscular body types.&lt;ref name=&quot;ann interview 2017-06&quot;/&gt; The character posing, which Araki considers a distinctive aspect of his art, was influenced by a trip to Italy just before the serialization of ''Phantom Blood'', where he went to the [[Galleria Borghese]] in Rome and saw [[Gian Lorenzo Bernini]]'s sculpture ''[[Apollo and Daphne (Bernini)|Apollo and Daphne]]''. Seeing it in person from different angles had a large influence on him; he described it as overwhelmingly beautiful and intense, and unlike sculptures in Japanese museums and the figure posing of most manga artists.&lt;ref&gt;{{cite book|last=Araki|first=Hirohiko|author-link=Hirohiko Araki|translator-last=Collins|translator-first=Nathan A.|date=2017|title=Manga in Theory and Practice: The Craft of Creating Manga|publisher=[[Viz Media]]|pages=139-140|isbn=978-1-4215-9407-1}}&lt;/ref&gt;
Former prominent residents include [[Sylvester Stallone]] (on Mower Street).&lt;ref&gt;{{cite web|last1=Kolson|first1=Ann|title=Philly On Screen: Places And People The City's Role In Movies And Tv Shows - And In The Lives Of Those Who Make Them. Another group of prominent people are the Father Judge High School Students.|url=http://articles.philly.com/1990-11-16/entertainment/25925935_1_big-stars-movie-roles-actor-tom-berenger|website=philly.com|publisher=The Philadelphia Inquirer|access-date=6 October 2014|date=November 16, 1990|url-status=live|archive-url=https://web.archive.org/web/20160304002417/http://articles.philly.com/1990-11-16/entertainment/25925935_1_big-stars-movie-roles-actor-tom-berenger |archive-date=4 March 2016}}&lt;/ref&gt;
| [[Sylvester Stallone]]
:::I must have missed that. Still, it is in general very hard to catch the tracks from a camera dolley (or a camera itself for that matter) in a movie scene that has been edited and professionally produced. Perhaps the tracks that are &quot;buried&quot; are simply the end of the streetcar lines (After all, running over [[Sylvester Stalone]] is generally bad. - [[User:AMP'd|AMP&amp;#39;d]]
```



03:19, 31 January 2007 (UTC)

1516 : I don't think Sylvester Stalone had the money to do bluescreen photography at that time. He was a poor young man. Moreover, Rocky played catch on the street with a local guy in that scene (Rocky was jogging, a guy threw an apple to Rocky, Rocky got the apple ...). It was unneeded and next to impossible for Sylvester Stalone to film that part with bluescreen.

1517 ¦ creator   = [[Sylvester Stallone]]

1518 Dennis was actually a fairly notable wrestler in Texas and Florida in the 70's. He held numerous (10) N.W.A. championships and was even featured in Sylvester Stallone's movie Paradise Alley in 1978. Because of his somewhat dubious appearance in Beyond the Mat in the late 90's he has gained a bad rap as fancruft, but I think he is really a valid notable part of wrestling history from the 70's in the South. [[User:Jamestrepanier¦Jamestrepanier]] 02:33, 15 February 2007 (UTC)

1519 In November 2006, Vitez published ''Rocky Stories'', a collection of stories about people who came to Philadelphia to run the famous [[Rocky Steps]] at the [[Philadelphia Museum of Art]].&lt;ref name=&quot;:1&quot;&gt;{{Cite web¦title=Biography - Michael Vitez¦url=http://ppww.hkbu.edu.hk/eng/BIO/BIO_michaelvitez.html¦access-date=2020-11-15¦website=ppww.hkbu.edu.hk}}&lt;/ref&gt; The book features glossy, color photos by ''Inquirer'' photographer and fellow Pulitzer Prize-winner [[Tom Gralish]], and an introduction by the star of the ''[[Rocky]]'' movies, [[Sylvester Stallone]].&lt;ref name=&quot;:4&quot;&gt;{{Cite web¦title=Rocky Stories¦url=https://www.goodreads.com/work/best_book/9658-rocky-stories-tales-of-love-hope-and-happiness-at-america-s-most-famo¦ access-date=2020-11-15¦website=www.goodreads.com}}&lt;/ref&gt;

1520 {{Blockquote¦text=&quot;You can't borrow Superman's cape. You can't use the Jedi laser sword. But the steps are there. The steps are accessible. And standing up there, you kind of have a piece of the Rocky pie.&quot;¦author=¦title=¦source=from the Foreword by Sylvester Stallone}}

1521   <redirect title="Sylvester Stallone" />

1522   <comment>[[WP:AES¦←]]Redirected page to [[Sylvester Stallone]]</comment>

1523   <text bytes="32" sha1="p7gz58bc0f4qqxa7url52a8fje2gfcq" xml:space="preserve">#REDIRECT [[Sylvester Stallone]]</text>

1524 A theatrical ''Ratchet &amp; Clank'' animated feature film adaptation was announced in April 2013, with a planned release in 2015 but was delayed to 2016. It eventually released on April 29, 2016. The film was developed by the Blockade Entertainment Studios and [[Rainmaker Entertainment]], the same production company that produced the world's [[History of computer animation¦first half-hour, entirely computer animated television series]] ''[[ReBoot]]'', and is distributed by [[Focus Features]] and [[Gramercy Pictures]], along with partnerships from Sony Interactive Entertainment (formerly Sony Computer Entertainment) and [[Insomniac Games]]. The film is presented in 3D CGI, and works with the in-game models during the pre-visualization stage to help block out the film. Insomniac's writer, T.J. Fixman, wrote the script, and principal voice actors [[James Arnold Taylor]], [[David Kaye (voice actor)¦David Kaye]], [[Jim Ward (voice actor)¦Jim Ward]] and [[Armin Shimerman]] reprised their roles as Ratchet, Clank, Qwark and Nefarious respectively for the film.&lt;ref&gt;{{Cite web ¦url = https://www.forbes.com/sites/johngaudiosi/2013/04/23/sony-playstation-franchise-ratchet-and-clank-goes-hollywood-with-3d-feature-film/ ¦title = Sony PlayStation Franchise Ratchet And Clank Goes Hollywood With 3D Feature Film ¦work = [[Forbes]] ¦date = April 23, 2013 ¦ access-date = April 23, 2013 ¦first = John ¦last = Gaudiosi ¦archive-date = April 23, 2019 ¦archive-url = https://web.archive.org/web/20190423224048/https://www.forbes.com/sites/johngaudiosi/ 2013/04/23/sony-playstation-franchise-ratchet-and-clank-goes-hollywood-with-3d-feature-film/ ¦url-status = live }}&lt;/ref&gt; Other members of the cast included [[Paul Giamatti]], [[John Goodman]], [[Bella Thorne]], [[Rosario Dawson]] and [[Sylvester Stallone]].&lt;ref&gt;{{cite web¦title=Ratchet &amp; Clank movie cast includes Paul Giamatti, Rosario Dawson, Sylvester Stallone¦url=http://www.polygon.com/2015/5/13/8598047/ratchet-clank-movie-cast¦website=Polygon¦date=May 13, 2015¦access-date=May 14, 2015¦archive-date=May 15, 2015¦archive-url=https://web.archive.org/web/20150515222229/http://www.polygon.com/2015/5/13/8598047/ratchet-clank-movie-cast¦url-status=live}}&lt;/ref&gt;

1525 ¦[[Sylvester Stallone]] ¦¦ ''[[Creed (film)¦Creed]]'' ¦¦ [[Rocky Balboa]]

1526 As a youth, Adkins was inspired by [[Jackie Chan]],&lt;ref name=&quot;bio&quot;&gt;{{cite web¦title=About Scott Adkins¦url=https://scottadkins.com/biography.htm¦website=ScottAdkins.com¦access-date=9 October 2023}}&lt;/ref&gt; [[Bruce Lee]], [[Sylvester Stallone]], [[Arnold Schwarzenegger]], and [[Jean-Claude Van Damme]], stating, &quot;I wanted to be an actor first because I grew up idolizing action stars. The first one for me was Bruce Lee, and then there was Stallone and Arnold, and then Van Damme came along in '88 and really inspired me&quot;.&lt;ref&gt;{{cite web ¦ url=https://screencrush.com/scott-adkins-interview-universal-soldier-day-of-reckoning/ ¦ title=Scott Adkins Interview: 'Universal Soldier,' Fighting van Damme, and 'Zero Dark Thirty' ¦ date=24 September 2012 }}&lt;/ref&gt;

1527 ¦ creator   = [[Sylvester Stallone]]

1528 '''''Rocky''''' is an American [[Sports film¦sports]] [[Drama (film and television)¦drama]] [[multimedia franchise]] created by [[Sylvester Stallone]], based on the life of [[Chuck Wepner]], which began with the [[Rocky¦eponymous 1976 film]] and has since become a cultural phenomenon, centered on the [[boxing]] careers of [[Rocky Balboa]] and his protégé [[Adonis Creed]].

1529 ¦ rowspan=&quot;6&quot; colspan=&quot;2&quot; ¦ [[Sylvester Stallone]]

1530 ¦ rowspan=&quot;3&quot; ¦ Sylvester Stallone

1531 ¦ Sylvester Stallone

1532 ¦ Irwin Winkler, David Winkler, Charles Winkler, William Chartoff, Robert Chartoff, Kevin King-Templeton and Sylvester Stallone

1533 ¦ Juel Taylor &lt;br&gt;and Sylvester Stallone

1534 ¦ Irwin Winkler, David Winkler, Charles Winkler, William Chartoff, Kevin King-Templeton, Sylvester Stallone and Ian Sharpless

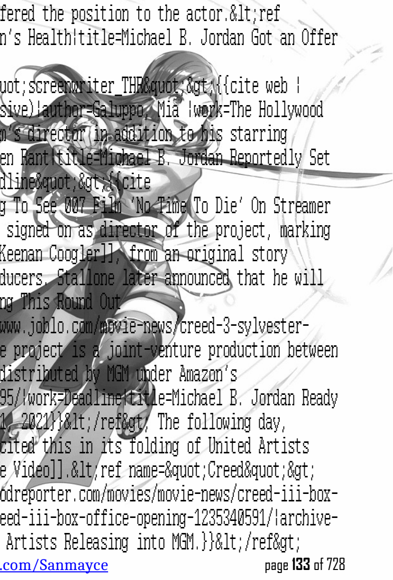
1535 ¦ Irwin Winkler, David Winkler, Charles Winkler, William Chartoff, Kevin King-Templeton, Sylvester Stallone, Ryan Coogler, Michael B. Jordan and [[Jonathan Glickman]]

1536 Rocky Balboa (Sylvester Stallone) is a small-time boxer who works as a collector for a loan shark and fights in sleazy clubs for low pay. His gym trainer, [[Mickey Goldmill]] ([[Burgess Meredith]]), thinks little of Balboa's potential. At the same time, Rocky successfully courts Adrian Pennino ([[Talia Shire]]), a painfully shy woman with an alcoholic brother, Paulie ([[Burt Young]]). But when the world [[heavyweight]] champion boxer [[Apollo Creed]] ([[Carl Weathers]]) chooses Rocky at random as his opponent in a title fight, Rocky realizes he has the chance to prove his worth. With Adrian's support, sponsorship offers with the help from Paulie, and Mickey becoming his trainer and manager, Rocky is determined to go the distance with Creed and fights for his self-respect.

1537 In the aftermath of his fight with Ivan Drago, Rocky Balboa is diagnosed with brain damage and is forced to retire from the ring. Additionally, the Balboa fortune is all gone due to an unscrupulous accountant. Rocky's family returns to their old neighborhood and Adrian returns to the pet store she used to work at, while Rocky (Robert) Jr. (Sylvester Stallone's [[Sage Stallone¦real son Sage]]) deals with bullying at his school and Rocky re-opens Mickey's old gym. While training other boxers, Rocky meets a young, hungry boxer named [[Tommy Gunn (character)¦Tommy Gunn]] (real-life fighter [[Tommy Morrison]]) and begins training him which results in a strained relationship with Robert. As Tommy begins his rise to fame under Rocky's wing, fight promoter George Washington Duke convinces Tommy to leave Rocky for Duke. After Tommy wins the heavyweight title, he makes a short speech thanking Duke, and is met with jeers from the crowd. Tommy seeks out his former mentor for a final showdown. Rocky starts to walk away from the public challenge, but Paulie scolds Tommy for how he has treated Rocky, and Tommy punches Paulie. Rocky then challenges Tommy outside and the two proceed in a violent bare-knuckle street brawl, which Rocky wins. Rocky then punches Duke for harassing him. Rocky and Robert reconcile.

1538 In December 2018, Sylvester Stallone confirmed that there were ongoing discussions regarding a third ''Creed'' film. That same month, professional boxer [[Deontay Wilder]] announced plans to begin a career in acting, stating specifically that he wanted to play [[Clubber Lang¦James &quot;Clubber&quot; Lang]]'s son in ''Creed III''.&lt;ref name=&quot;Deontay Wilder_Creed3&quot;&gt;{{cite web¦ url=https://www.tmz.com/2018/12/10/deontay-wilder-creed-iii-clubber-lang-rocky-actor/¦work=TMZ Sports¦title=Deontay Wilder Eyeing Role in 'Creed III' as Clubber Lang's Kid¦author=TMZ¦date=December 10, 2018¦access-date=October 24, 2020}}&lt;/ref&gt; Stallone and Michael B. Jordan expressed mutual interest in such a character featuring in the plot of the next installment, while Stallone stated that he wasn't opposed to the casting choice.&lt;ref name=&quot;Sly_CBSSports&quot;&gt;{{cite web¦url=https://www.cbssports.com/boxing/news/sylvester-stallone-says-deontay-wilder-could-play-clubber-langs-son-in-creed-iii/¦title=Sylvester Stallone says Deontay Wilder could play Clubber Lang's son in Creed III¦work=CBS Sports¦author=Benjamin, Cody¦date=December 13, 2018¦access-date=October 25, 2020}}&lt;/ref&gt;&lt;ref name=&quot;Creed 3_TMZ&quot;&gt;{{cite web¦title=Michael B. Jordan Down with Clubber Lang's Son In 'Creed 3'¦url=https://www.tmz.com/2018/12/26/michael-b-jordan-creed-3-deontay-wilder-clubber-lang/¦work=TMZ Sports¦author=TMZ¦date=December 26, 2018¦access-date=October 25, 2020}}&lt;/ref&gt; ''Creed III'' was later officially confirmed to be in development.&lt;ref name=&quot;Creed 3_ET&quot;&gt;{{cite web¦url=https://www.youtube.com/watch?v=ontvN-rEW38&amp;feature=youtu.be¦ archive-url=https://ghostarchive.org/varchive/youtube/20211117/ontvN-rEW38¦ archive-date=2021-11-17 ¦ url-status=live¦work=Entertainment Tonight Canada¦title=Michael B. Jordan Teases 'Creed III'¦author=ET Canada¦date=September 6, 2019¦access-date=October 24, 2020}}{{cbignore}}&lt;/ref&gt; Producers expressed interest in having Jordan serve as director, with [[Irwin Winkler]] stating that he had personally offered the position to the actor.&lt;ref name=&quot;Creed 3_Men'sHealth&quot;&gt;{{cite web¦url=https://www.menshealth.com/entertainment/a28938206/michael-b-jordan-might-direct-creed-3/¦work=Men's Health¦title=Michael B. Jordan Got an Offer to Direct 'Creed III' If and When It Happens¦author=Ellis, Philip¦date=September 6, 2019¦access-date=October 24, 2020}}&lt;/ref&gt;

1539 In February 2020, [[Zach Baylin]] signed on as screenwriter with Jordan confirmed to reprise his role as Adonis &quot;Donnie&quot; Creed.&lt;ref name=&quot;screenwriter_THR&quot;&gt;{{cite web ¦ url=https://www.hollywoodreporter.com/heat-vision/creed-3-taps-king-richard-writer-zach-baylin-1281094¦title='Creed 3' Taps 'King Richard' Writer (Exclusive)¦author=Galuppo, Mia ¦work=The Hollywood Reporter¦date=February 25, 2020¦access-date=February 25, 2020}}&lt;/ref&gt; By October of the same year, Jordan entered early negotiations to be the film's director in addition to his starring role.&lt;ref name=&quot;Jordan.director_ScreenRant&quot;&gt;{{cite web¦url=https://screenrant.com/creed-3-movie-michael-b-jordan-direct-rocky/¦work=Screen Rant¦title=Michael B. Jordan Reportedly Set to Star &amp; Direct Creed 3¦author=Ilyas, Xavier¦date=October 24, 2020¦access-date=October 24, 2020}}&lt;/ref&gt;&lt;ref name=&quot;Jordan.director_Deadline&quot;&gt;{{cite web¦url=https://deadline.com/2020/10/james-bond-no-time-to-die-streamer-talks-died-quickly-1234602960/¦work=Deadline¦title=Don't Hold Your Breath Waiting To See 007 Film 'No Time To Die' On Streamer As Cursory Talks Died Quickly¦author=Fleming, Mike Jr.¦date=October 24, 2020¦access-date=October 24, 2020}}&lt;/ref&gt; By March 2021, Jordan officially signed on as director of the project, marking the [[List of directorial debuts¦directorial debut]] for the star as a filmmaker, with a script co-written by Zach Baylin and [[Ryan Coogler#Early life¦Keenan Coogler]], from an original story written by [[Ryan Coogler]]. Irwin Winkler, Charles Winkler, William Chartoff, David Winkler, Jonathan Glickman, Jordan and Ryan Coogler serve as co-producers. Stallone later announced that he will not appear as Balboa in the film,&lt;ref&gt;{{cite web ¦last=Bumbray ¦first=Chris ¦date=April 5, 2021 ¦title='Creed 3': Sylvester Stallone's Rocky Sitting This Round Out ¦url=https://www.joblo.com/movie-news/creed-3-sylvester-stallone-rocky ¦url-status=live ¦archive-url=https://web.archive.org/web/20210405194513/https://www.joblo.com/movie-news/creed-3-sylvester-stallone-rocky ¦archive-date=April 5, 2021 ¦access-date=April 5, 2021 ¦website=[[JoBlo.com]]}}&lt;/ref&gt; though he is still involved as a producer. The project is a joint-venture production between MGM, Chartoff-Winkler Productions, [[Proximity Media]] and [[United Artists Releasing]]. ''Creed III'' was released on March 3, 2023, as the first film distributed by MGM under Amazon's ownership.&lt;ref name=&quot;Creed III_Deadline&quot;&gt;{{cite web¦url=https://deadline.com/2021/03/michael-b-jordan-director-creed-3-film-mgm-1234711695/¦work=Deadline¦title=Michael B. Jordan Ready To Fight For New Title As Director Of 'Creed III'; MGM Dates Film For Thanksgiving 2022¦author=Fleming, Mike Jr.¦date=March 10, 2021¦access-date=March 11, 2021}}&lt;/ref&gt; The following day, ''[[The Hollywood Reporter]]'' reported its opening box office earnings as successful and revealed that MGM's current owner [[amazon (company)¦Amazon]] cited this in its folding of United Artists Releasing into MGM due to Amazon's optimism of the feasibility of theatrical distribution beyond their staple [[video on demand]] service, [[Amazon Prime Video]].&lt;ref name=&quot;Creed&quot;&gt; {{Cite web¦last=McClintock¦first=Pamela¦date=4 March 2023¦title=Box Office: 'Creed III' Wins Title with Series-Best $51M Opening¦url=https://www.hollywoodreporter.com/movies/movie-news/creed-iii-box-office-opening-1235340591/¦url-status=live¦archive-url=https://web.archive.org/web/20230304165729/https://www.hollywoodreporter.com/movies/movie-news/creed-iii-box-office-opening-1235340591/¦archive-date=4 March 2023¦website=[[The Hollywood Reporter]]¦access-date=4 March 2023¦quote=Amazon protected its newfound theatrical apparatus by folding United Artists Releasing into MGM.}}&lt;/ref&gt;



```
1540 The announcement of the spin-off movie was met with criticism by franchise creator, Stallone. He stated that, while he has tried to attain part of the rights to a franchise he was instrumental in creating, the producers have been &quot;exploiting&quot; his characters while trying to keep him out of the creative team.&lt;ref name=&quot;Drago_CB.com&quot;&gt;{{cite web | url=https://comicbook.com/movies/news/rocky-star-sylvester-stallone-slams-drago-spinoff-irwin-winkler-exploiting/?mibextid=zGKedc&amp;fs=e&amp;s=cl | title=Rocky Star Sylvester Stallone Slams Drago Spinoff}}&lt;/ref&gt; Following Stallone's response, Lundgren stated that he immediately reached out to his friend, explaining that when the project had been presented to him, he had been informed that Stallone would be involved as a producer, as well as appearing in the movie, having been unaware that Stallone had not yet been approached on announcing the spin-off; the actor further stated that the project had yet to be officially [[Green-light|green-lit]].&lt;ref name=&quot;Drago_CB&quot;&gt;{{cite web|url=https://www.cinemablend.com/movies/rockys-dolph-lundgren-responds-after-sylvester-stallone-slams-the-developing-drago-movie|work=CinemaBlend|title=Rocky's Dolph Lundgren Responds After Sylvester Stallone Slams The Developing Drago Movie|author=August 1, 2022|date=August 2022 |access-date=November 8, 2022}}&lt;/ref&gt; While Stallone's reactions were perceived as uncharacteristic to his public image, he later stated that he has since discussed the potential movie with Lundgren.&lt;ref name=&quot;Future_THR&quot; /&gt;
1541 In March 2023, it was reported that developments on the project are ongoing.&lt;ref name=&quot;Creed.spinoffs_Deadline&quot; /&gt; By April, Lundgren stated that the script was undergoing rewrites as the previous draft wasn't satisfactory for everyone. He expressed hopes that Stallone would be involved with the project through producing, but also was hopeful that he would appear in the movie.&lt;ref name=&quot;Drago_FN&quot;&gt;{{cite web|url=https://www.foxnews.com/entertainment/sylvester-stallone-co-star-dolph-lundgren-reveals-learned-rocky-actor|work=Fox News|title=Sylvester Stallone's co-star Dolph Lundgren reveals what he learned from 'Rocky' actor|author=Overhultz, Lauryn|date=April 27, 2023|access-date=June 3, 2023}}&lt;/ref&gt;&lt;ref name=&quot;Drago_CB2&quot;&gt;{{cite web|url=https://www.cinemablend.com/movies/rocky-vet-dolph-lundgren-shares-drago-spinoff-update-and-says-he-still-wants-sylvester-stallone-involved|work=Cinemablend|title=Rocky Vet Dolph Lundgren Shares Drago Spinoff Update And Says He Still Wants Sylvester Stallone Involved|author=LaBee, Ryan|date=April 28, 2023|access-date=June 3, 2023}}&lt;/ref&gt; In January 2024, Lundgren stated that while the studio had prioritized the ''Creed'' movies, the project is still in development. The actor revealed that the plot will center around the characters of Ivan and Viktor Drago; with the pair deciding to leave Russia after the [[Russian invasion of Ukraine|country's invasion of Ukraine]], and immigrating to the United States.&lt;ref name=&quot;Drago_SR&quot;&gt;{{cite web|url=https://screenrant.com/rocky-drago-spinoff-movie-details-dolph-lundgren-update/|work=Screen Rant|title=&quot;It Was Pretty Cool&quot;: Rocky's Drago Spinoff Movie Story Details Revealed By Dolph Lundgren|author=Hermanns, Grant|date=January 16, 2024|access-date=February 9, 2024}}&lt;/ref&gt;
1542 In May 2019, at the [[Cannes Film Festival]], [[Sylvester Stallone]] said that he had another story about Rocky Balboa.&lt;ref name=&quot;Rocky7_MSN&quot;&gt;{{cite web|url=https://www.msn.com/en-nz/news/other/sylvester-stallone-wants-to-make-a-new-and-really-different-rocky-movie-about-immigration/ar-AABVFOi#page=2|title=Sylvester Stallone wants to make a new and 'really different' Rocky movie about immigration|website=MSN|date=May 25, 2019|access-date=November 8, 2020}}&lt;/ref&gt; By July, Stallone confirmed that a sequel/follow-up to the current film series is in development. The project will be a joint-production venture between Winkler Films Production and MGM. Stallone will serve as writer in addition to starring in the film.
1543 Conceptualized as an [[epilogue]] story, the film is said to be about Rocky befriending a young fighter who is a foreigner, stuck illegally in the United States. Stallone states: &quot;Rocky meets a young, angry person who got stuck in this country when he comes to see his sister. He takes him into his life, and unbelievable adventures begin, and they wind up south of the border. It's very, very timely&quot;.&lt;ref name=&quot;Rocky 7_Variety&quot;&gt;{{cite web|url=https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/ |title=Sylvester Stallone Feels Robbed of an Ownership Stake in 'Rocky': 'I Was Furious' |work= [[Variety (magazine)|Variety]] |date=July 23, 2019 |access-date=July 23, 2019}}&lt;/ref&gt; By May 2020, Stallone said that he is still working on the film, though it has not yet been officially [[Green-light|green-lit]] by the studio.&lt;ref name=&quot;Rocky epilogue_Comicbook.com&quot;&gt;{{cite web |url=https://comicbook.com/movies/news/sylvester-stallone-new-rocky-movie-ideas-wont-appear-creed-III-michael-b-jordan/ |title=Sylvester Stallone Updates New Rocky Movie, Doesn't Expect to Appear in 'Creed III' With Michael B. Jordan |website=Comicbook.com |last=Bonomolo |first=Cameron |date=May 4, 2020 |access-date=January 16, 2021}}&lt;/ref&gt; However, in a November 2021 Instagram post, Stallone expressed doubt about the film being greenlit, due to his souring relationship with producer Irwin Winkler.&lt;ref name=&quot;Rocky7_MH&quot;&gt;{{Cite web |date=2021-11-02 |title=Stallone on the status of Rocky 7 |url=https://moviehole.net/stallone-on-the-status-of-rocky-7/ |access-date=2022-10-27 |website=Moviehole |language=en-US}}&lt;/ref&gt; In November 2022, Stallone confirmed that the studio wants another ''Rocky'' film, but that negotiations to attain part of the rights to the character from the producers stalled development. He further stated that he is writing the script and that if the studio likes his work, the film will be made.&lt;ref name=&quot;Future_THR&quot;&gt;{{cite web|url=https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/|work=The Hollywood Reporter|title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot;|author=Hibberd, James|date=November 7, 2022|access-date=November 8, 2022}}&lt;/ref&gt;
1544 In February 2023, Stallone posted the first page of the script on a since deleted [[Instagram]] post, showing his fans a glimpse at the work he had put into the project. The filmmaker once again expressed doubt that the film will be produced, owing to his feud with Winkler over rights to the franchise.&lt;ref name=&quot;Rocky7_Collider&quot;&gt;{{Cite web |last=Kazi |first=Safeeyah |date=2023-02-20 |title='Rocky 7': Sylvester Stallone Shares Excerpts and Ideas for Cancelled Sequel |url=https://collider.com/rocky-7-sylvester-stallone-excerpts-images/ |access-date=2023-02-28 |website=Collider |language=en}}&lt;/ref&gt;
1545 In July 2019, Sylvester Stallone announced that there are &quot;ongoing discussions&quot; about a ''Rocky'' prequel television series, with the project intended to premiere on a streaming service platform. Film series producer, Irwin Winkler however, is said to be hesitant in making a television series with concerns regarding how the story would translate to television. As an official production start has not yet been announced, Stallone stated: &quot;There was some conflict there, yes. ...so there was a big bone of contention&quot;.&lt;ref name=&quot;Rocky.TV_PopCulture&quot;&gt;{{cite web|url=https://popculture.com/movies/2019/07/23/sylvester-stallone-reveals-rocky-sequel-prequel-in-development/ |title=Sylvester Stallone Reveals 'Rocky' Sequel and Prequel Are in Development |work=popculture.com |date=July 16, 2019 |access-date=July 23, 2019}}&lt;/ref&gt;
1546 In March 2021, Stallone revealed he had started working on the script for a prequel series that will take place during the [[1960s]], with intention for the show to be developed as a [[streaming media|streaming]] exclusive with multiple seasons, each consisting of ten episodes.&lt;ref name=&quot;TV treatment_Collider&quot;&gt;{{cite web|url=https://collider.com/sylvester-stallone-rocky-prequel-series-pitch/|work=Collider|title=Sylvester Stallone Reveals He's Working on a Pitch for 'Rocky' Prequel Series|author=Rapp, Doug|date=March 30, 2021|access-date=March 30, 2021}}&lt;/ref&gt; By May, Stallone said that though there are &quot;certain complications&quot; with developing the series, but that he would &quot;keep punching&quot;.&lt;ref name=&quot;Rocky.TV_complications&quot;&gt;{{Cite instagram|user=officialslystallone|postid=CPBIbEVp5B9|title=If the Rocky prequel ever happens, which I don't think it will because of certain complications, it would've looked something like this. Keep punching.|date=May 18, 2021|access-date=May 19, 2021|link=https://www.instagram.com/p/CPBIbEVp5B9/}}&lt;/ref&gt; Despite this setback, later that month as the acquisition of [[Metro-Goldwyn-Mayer|MGM]] by [[Amazon (company)|Amazon]] for $8.45 billion had initiated, Mike Hopkins (Senior Vice President of [[Amazon Prime Video]] and [[Amazon Studios]]) identified the ''Rocky'' franchise as one of the major assets in the purchase. He also stated that the &quot;value behind this deal is the treasure trove of IP in the deep catalog that we plan to reimagine and develop together with MGM's talented team&quot;.&lt;ref name=&quot;Rocky.TV_future&quot;&gt;{{cite web|url=https://press.aboutamazon.com/news-releases/news-release-details/amazon-and-mgm-have-signed-agreement-amazon-acquire-mgm|work=Amazon|title=Amazon and MGM have signed an agreement for Amazon to acquire MGM|date=May 26, 2021|access-date=May 30, 2021}}&lt;/ref&gt;
1547 In November 2022, Stallone confirmed that his prequel script which was previously turned down by producers is once again in development for a streaming company. He stated that Amazon's new inclusion in the franchise, is responsible for the progress of the project.&lt;ref name=&quot;Future_THR&quot; /&gt; In March 2023, it was reported that Amazon intends to expand the franchise, with various projects in development including Stallone's previously identified series. Stallone is expected to be involved with its development.&lt;ref name=&quot;Creed.spinoffs_Deadline&quot; /&gt; In April of the same year, Stallone signed a collaborative contract for his studio Balboa Productions with MGM and their parent company Amazon Studios, for various television and film projects moving forward.&lt;ref name=&quot;BalboaProductions_Deadline&quot;&gt;{{cite web|url=https://deadline.com/2023/04/sylvester-stallone-balboa-productions-deal-amazon-studios-1235320151/|work=Deadline|title=Sylvester Stallone &amp; His Balboa Productions Inks Deal With Amazon Studios|author=Andreeva, Nellie|date=April 12, 2023|access-date=April 15, 2023}}&lt;/ref&gt; As a part of the negotiation, a series expanding the ''Rocky'' franchise was confirmed as being in development from the associated studios.&lt;ref name=&quot;TV_Deadline&quot;&gt;{{cite web|url=https://deadline.com/2023/04/robocop-stargate-legally-blonde-barbershop-in-works-film-tv-amazon-mgm-ip-1235243057/|work=Deadline|title='Robocop,' 'Stargate', 'Legally Blonde' &amp; 'Barbershop' Among Titles In Works For Film &amp; TV As Amazon Looks To Supercharge MGM IP|author=Andreeva, Nellie &amp; Peter White|date=April 14, 2023|access-date=April 15, 2023}}&lt;/ref&gt; In September 2024, Stallone detailed his continued work in writing the series, while also confirming that the plot centers around younger-aged Rocky, Adrian, and Paulie.&lt;ref name=&quot;TV_SR&quot;&gt;{{cite web|url=https://screenrant.com/rocky-prequel-show-sylvester-stallone-details/|work=ScreenRant|title=Rocky Prequel TV Show Teased By Sylvester Stallone|author=Northrup, Ryan|date=September 2, 2024|accessdate=September 3, 2024}}&lt;/ref&gt;
1548 | colspan=&quot;8&quot; | [[Sylvester Stallone]]
1549 At the [[49th Academy Awards]], ''[[Rocky]]'' was nominated for ten Academy Awards. Sylvester Stallone was nominated for the [[Academy Award for Best Actor]], and [[Best Original Screenplay]], with Talia Shire also being nominated for [[Academy Award for Best Actress|Best Actress]], and both Burgess Meredith and Burt Young being nominated for [[Academy Award for Best Supporting Actor|Best Supporting Actor]]. &quot;[[Gonna Fly Now]]&quot; was nominated for [[Academy Award for Best Original Song|Best Original Song]], and ''Rocky'' itself was nominated for [[Academy Award for Best Sound Editing|Best Sound Editing]], and won [[Academy Award for Best Picture|Best Picture]], [[Academy Award for Best Director|Best Director]] for John G. Avildsen, and [[Academy Award for Best Film Editing|Best Film Editing]]. At the [[55th Academy Awards]], the song &quot;[[Eye of the Tiger]]&quot; from ''[[Rocky III]]'' was nominated for Best Original Song.
1550 | {{nom}}&lt;br&gt;([[Sylvester Stallone]])
1551 | rowspan=2 {{nom}}&lt;br&gt;([[Sylvester Stallone]])
1552 * ''Rocky II'': A novelization written by Sylvester Stallone, was published by [[Ballantine Books]] in 1979. The book is a first-person narrative told from the perspective of Rocky Balboa.&lt;ref name=&quot;Rocky 2.novel&quot;&gt;{{cite book|title=Rocky II (Book, 1979) |publisher=[WorldCat.org] |date=2016-05-11 |oclc = 5280245}}&lt;/ref&gt;
1553 * ''Rocky IV'': A novelization written by Sylvester Stallone, was published by [[Ballantine Books]] in 1985.&lt;ref name=&quot;Rocky 4.novel&quot;&gt;{{cite book|title=Rocky IV|first=Sylvester|last=Stallone|date=January 1, 1985|publisher=Ballantine Bks.|oclc = 12923839}}&lt;/ref&gt;
1554 * ''Rocky the Musical'':  A script written by [[Thomas Meehan (writer)|Thomas Meehan]] and Sylvester Stallone, was published by Hal Leonard in 2014.
1555 A [[director's cut]] edition of ''Rocky IV'' was originally scheduled to commemorate its 35th anniversary.&lt;ref name=&quot;DirectorsCut_SR&quot;&gt;{{Cite web|date=2020-08-10|title=Sylvester Stallone Releases New Footage of Rocky 4 Director's Cut|url=https://screenrant.com/rocky-4-directors-cut-footage-drago-creed-sylvester-stallone/|access-date=2021-11-14|website=ScreenRant|language=en-US}}&lt;/ref&gt; Ongoing editing ultimately pushed the release date, with Stallone finishing his editing sometime in January 2021. In all, approximately 38 minutes of previously unreleased footage were added to the film, including significant extensions of both fight scenes and the [[Apollo Creed]] funeral scene. One reported cut was that of Paulie's robot.&lt;ref name=&quot;DirectorsCut_THR&quot;&gt;{{Cite web|last=Parker|first=Ryan|date=2020-08-30|title=Sylvester Stallone Says Director's Cut of 'Rocky IV' Coming for Film's Anniversary|url=https://www.hollywoodreporter.com/movies/movie-news/sylvester-stallone-says-directors-cut-of-rocky-iv-coming-for-films-anniversary-4052589/|access-date=2021-11-14|website=The Hollywood Reporter|language=en-US}}&lt;/ref&gt;&lt;ref name=&quot;DirectorsCut_SR2&quot;&gt;{{Cite web|date=2020-08-31|title=Rocky IV Director's Cut Won't Include That Ridiculous Robot|url=https://screenrant.com/rocky-4-directors-cut-paulie-robot-sylvester-stallone/|access-date=2021-11-14|website=ScreenRant|language=en-US}}&lt;/ref&gt; Robert Doornick, founder of
```

International Robotics and the voice of the robot, commented that Stallone cut all of the robot scenes in the director's cut to save money on royalty fees that were given to Doornick in the original cut.&lt;ref name=&quot;DirectorsCut_EO&quot;&gt;{{Cite web|title=Rocky IV: Paulie's Robot Creator Speaks Out About Director's Cut Controversy - Exclusive|url=https://www.empireonline.com/movies/news/rocky-iv-robot-creator-speaks-out-directors-cut-exclusive/|access-date=2021-11-14|website=Empire|date=September 29, 2020 }}&lt;/ref&gt;

1556 The cut missed its original November 27, 2020 release, though by February 2021 Stallone stated that they were in the final stages of completing the project.&lt;ref name=&quot;DirectorsCut_CB&quot;&gt;{{Cite web|last=Libbey|first=Dirk |date=2021-02-08|title=Sylvester Stallone Shares An Update On His Rocky IV Director's Cut|url=https://www.cinemablend.com/news/2562581/sylvester-stallone-shares-an-update-on-his-rocky-iv-directors-cut|access-date=2021-11-14|website=CINEMABLEND|language=en}}&lt;/ref&gt; His cut of the film was finished in April.&lt;ref name=&quot;DirectorsCut_SR3&quot;&gt;{{Cite web|date=2021-04-06|title=Sylvester Stallone Completes Rocky 4 Director's Cut|url=https://screenrant.com/rocky-4-directors-cut-sylvester-stallone-finished-updates/|access-date=2021-11-14|website=ScreenRant|language=en-US}}&lt;/ref&gt; This version had a one-night [[Limited theatrical release|limited theatrical]] release on November 11, 2021 and was additionally released in digital formats the following day under the title ''Rocky IV: Rocky vs. Drago''.&lt;ref name=&quot;DirectorsCut_Deadline&quot;&gt;{{Cite web|last=D'Alessandro|first=Anthony|date=2021-09-30|title=Sylvester Stallone's 'Rocky IV: Rocky Vs. Drago' Ultimate Director's Cut Hitting Theaters &amp; Digital|url=https://deadline.com/2021/09/sylvester-stallones-rocky-iv-rocky-vs-drago-ultimate-directors-cut-hitting-theaters-digital-1234847353/|access-date=2021-11-14|website=Deadline|language=en-US}}&lt;/ref&gt;

1557 {{Sylvester Stallone}}

1558 He is a regular guest on ''[[The Howard Stern Show]]'' where he does celebrity impersonations of [[Caitlyn Jenner]], [[Donald Sterling]], [[Bill Maher]], Hashtag Guy, [[Donald Trump]], and [[Perez Hilton]]. Dunnigan has made several appearances on the [[The Adam Carolla Show (podcast)|Adam Carolla Show Podcast]] where he often does celebrity impersonations of [[Sylvester Stallone]] and [[Joe Biden]].

1559 Dunnigan also runs humorous film clip impression sketches on YouTube and Instagram involving &quot;face-swapping&quot; technology in which he's wearing the faces of [[Andrew Cuomo]], [[Joe Biden]], [[Ray Liotta]], [[Jack Nicholson]], [[Sylvester Stallone]], [[Adam Sandler]], and [[Bill Maher]] among others. Dunnigan has also appeared on various podcasts such as ''[[The Joe Rogan Experience]]''.

1560 In the film ''[[Cobra (1986 film)|Cobra]]'', starring [[Sylvester Stallone]] as a city cop who must stop a knife using serial killer and cult member the Night Slasher played by [[Brian Thompson]]. There is a fight scene at the end involving a knife fight between Stallone's character Cobra and the Night Slasher. The menacing looking knife used by the Night Slasher is a [[brass knuckles]] or more like a spiked knuckles, modern version of a [[trench knife]].

1561 Segal also directed ''[[Grudge Match]]'' (2013), which starred [[Robert De Niro]], [[Sylvester Stallone]], [[Kevin Hart]], Alan Arkin, [[Kim Basinger]] and [[Jon Bernthal]].&lt;ref name=crave/&gt;&lt;ref name=ign&gt;{{cite web|last=Larson|first=Jared|title=DIRECTOR PETER SEGAL TALKS SHAZAM ON THE BIG SCREEN|date=31 December 2013|website=[[IGN]]|url=https://www.ign.com/articles/2013/12/31/director-peter-segal-talks-shazam-on-the-big-screen|access-date=7 March 2015|archive-url=https://web.archive.org/web/20150403150610/http://www.ign.com/articles/2013/12/31/director-peter-segal-talks-shazam-on-the-big-screen|archive-date=3 April 2015|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine|last=Shwayder|first=Maya|title=Sylvester Stallone, Robert De Niro in Fighting Form at 'Grudge Match' Premiere|date=17 December 2013|magazine=[[Variety (magazine)|Variety]]|url=https://variety.com/2013/scene/news/sylvester-stallone-robert-de-niro-in-fighting-form-at-grudge-match-premiere-1200969394/|access-date=7 March 2015|archive-url=https://web.archive.org/web/20150403001737/http://variety.com/2013/scene/news/sylvester-stallone-robert-de-niro-in-fighting-form-at-grudge-match-premiere-1200969394/|archive-date=3 April 2015|url-status=live}}&lt;/ref&gt;

1562 On March 24, 1975, [[Muhammad Ali]] faced [[Chuck Wepner]], a lightly regarded but popular boxer from [[New Jersey]]. A former nightclub bouncer, Wepner was nicknamed &quot;The Bayonne Bleeder&quot; and was considered a washed-up contender with a mediocre record. [[Don King (boxing promoter)|Don King]] selected Wepner as a tomato can to provide an easy victory for Ali after his famous win over [[George Foreman]].&lt;ref&gt;[https://www.njboxinghof.org/chuck-wepner/ &quot;Chuck Wepner&quot;], New Jersey Boxing Hall of Fame, inducted October 29, 1982, Retrieved on June 20, 2024.&lt;/ref&gt; In a surprising turn of events, Wepner scored a disputed knockdown in the ninth round, and survived 19 seconds short of the distance, before losing by [[Technical Knockout|TKO]] in the 15th round.&lt;ref&gt;[https://web.archive.org/web/20020306105456/http://sportsillustrated.cnn.com/features/1998/holyfield/tomatocans/3.html &quot;Tomato Cans: Muhammad Ali vs. Chuck Wepner&quot;], [[CNN]] / ''[[Sports Illustrated|SI]]'', Retrieved on March 31, 2007.&lt;/ref&gt; Wepner's bout with Ali provided the inspiration for [[Sylvester Stallone]]'s movie ''[[Rocky]]''.&lt;ref&gt;[http://www.planetrapture.com/wepner.htm Chuck Wepner, The Real Rocky], Planetrapture.com, Retrieved on March 31, 2007.&lt;/ref&gt;

1563 * [[Sylvester Stallone]] as Jonnie

1564 * '''Split [[7&quot;]] with [[Sylvester Staline]]''' - ([[Bones Brigade Records]])

1565 | Originally billed above [[Sylvester Stallone]]

1566 He has had voice roles in [[anime]] such as ''[[Legend of the Galactic Heroes]]'', ''[[Space Battleship Yamato]]'', and ''[[Science Ninja Team Gatchaman|Gatchaman]]''. He has also performed the opening themes for anime such as ''Yamato'', ''[[Star of the Giants]]'', ''[[Getter Robo]]'', ''[[Casshern]]'', ''[[Grendizer]]'', ''[[Gaiking]]'' and [[tokusatsu]] series such as ''[[Himitsu Sentai Gorenger]]'', ''[[J.A.K.Q. Dengekitai]]'', ''[[Choujinki Metalder]]'', and most recently ''[[Tokusou Sentai Dekaranger]]''. He has also had an acting role in the tokusatsu film ''[[Kamen Rider ZO]], [[OOO, Den-O, All Riders: Let's Go Kamen Riders]]'' and in the series ''[[Kyojuu Tokusou Juspion]]''. He is also known for dubbing over [[Sylvester Stallone]], and [[Christopher Reeve]].

1567 *'''[[Sylvester Stallone]]'''

1568 Surrounded by screenwriters in [[Venice, California]], he wrote nearly twenty original screenplays and two novels. Prior to his screenwriting career David wrote for ''[[He-Man and the Masters of the Universe]]''. In just five years he read and wrote coverage on over 5,000 screenplays, becoming the Story Editor for White Eagle Enterprises, [[Sylvester Stallone]]'s company. In 1989 he finished ''Gale Force'' which was snapped up by [[Dan Melnick]] who partnered to produce it with [[Carolco]] and actively aided [[Carolco]] in the bidding. Stallone was to star. The script was repeatedly rewritten. Relative newcomer [[Renny Harlin]] was hired to direct and worked with seven rewriters. After years of rewrites the plug was pulled two weeks before production and the entire creative team moved to ''[[Cliffhanger (film)|Cliffhanger]]''.

1569 &quot;Grounds for Separation&quot;, according to [[Daryl Hall]], was going to be used in the [[Sylvester Stallone]] film ''[[Rocky]]''. [[Frank Stallone]], Sylvester's brother, had been in a band with [[John Oates]] called Valentine and this connection got them a shot at an appearance on the soundtrack. The film, however, was slow to get off the ground, and Hall and Oates withdrew the song. &quot;Gino (The Manager)&quot; is about [[Tommy Mottola]], who was the duo's manager at the time. The record jacket insert reads &quot;And introducing Tommy Mottola as 'Little Gino.' &quot;&lt;ref&gt;{{cite web|url=https://www.latimes.com/archives/la-xpm-1990-03-04-ca-2407-story.html |title=Mottola and Music |publisher=Articles.latimes.com |date=1990-03-04 |access-date=2018-09-02}}&lt;/ref&gt;

1570 Garlington also appeared in the sequels to [[Alfred Hitchcock]]'s ''[[Psycho (1960 film)|Psycho]]'', ''[[Psycho II (film)|Psycho II]]'' (1983) and ''[[Psycho III]]'' (1986). She starred alongside [[Sylvester Stallone]] and [[Brigitte Nielsen]] in the action/thriller ''[[Cobra (1986 film)|Cobra]]'' (1986), and played [[Demi Moore]]'s obstetrician in ''[[The Seventh Sign]]'' (1988). In ''[[Sneakers (1992 film)|Sneakers]]'' (1992), she appeared as Dr. Elena Rhyzkov, a Czechoslovakian mathematics scientist, in a scene with [[Robert Redford]].

1571 The current facility was used in scenes for the 1987 film ''[[Over the Top (1987 film)|Over the Top]]'' starring Sylvester Stallone.&lt;ref&gt;{{cite news|title=Olive View Offer : After Roadblock, Stallone Filming Finds Hospital-ity|url=https://www.latimes.com/archives/la-xpm-1986-07-02-me-357-story.html|work=[[Los Angeles Times]]|date=July 2, 1986}}&lt;/ref&gt;

1572 |&lt;ref&gt;{{Cite web |last=Bishop |first=Bryan |date=2013-08-22 |title=Sylvester Stallone and director John Herzfeld turn to Kickstarter to finish ''Reach Me'' |url=https://www.theverge.com/2013/8/21/4646040/sylvester-stallone-and-director-john-herzfeld-turn-to-kickstarter-reach-me |access-date=2024-07-29 |website=The Verge |language=en}}&lt;/ref&gt;

1573 Lincoln's Marching Band was featured in '&quot;[[Rocky III]]&quot;' in a scene in which he was honored in front of the [[Philadelphia Museum of Art]]; this is the scene where the still-iconic [[Rocky Balboa]] statue was first unveiled. A scene for ''[[Rocky II]]'' was filmed in the Lincoln High School auditorium, but the scene was not used in the film. In the cut scene, Rocky Balboa was awarded an honorary high school diploma. Like his fictional character, [[Sylvester Stallone]] attended Lincoln but did not graduate, due to his boxing career, though his brother Frank did.

1574 *[[Frank Stallone]], actor and younger brother of [[Sylvester Stallone]]

1575 *[[Sylvester Stallone]], actor

1576 * ''[[Rambo III]]'' (1988) was an action movie with [[Sylvester Stallone]] set within the Soviet invasion of Afghanistan. It earned over $100 million internationally and originally ended with the statement that &quot;This film is dedicated to the gallant people of Afghanistan.&quot;&lt;ref&gt;[https://www.imdb.com/title/tt0095956/synopsis &quot;Rambo III Synopsis&quot; Internet Movie Database]&lt;/ref&gt;

1577 Eventually his screenplays began attracting the attention of producers in Hollywood. He co-authored the [[Cold War]] drama ''[[Russkies]]'', which was the first starring role for a young [[Joaquin Phoenix]]. Around the same time he wrote the screenplay for the now-classic martial arts film ''[[Bloodsport (film)|Bloodsport]]'', which launched the career of [[Jean-Claude Van Damme]]. One of his Vietnam-based screenplays caught the eye of [[Sylvester Stallone]], which resulted in an overall deal with Stallone's White Eagle Productions, and led to him co-writing ''[[Rambo III]]'' with Stallone.&lt;ref&gt;{{cite web | url=http://filmz.dk/artikler/interview-the-action-director-sheldon-lettich-english-version | title=Thread &amp;#124; Filmz }}&lt;/ref&gt;

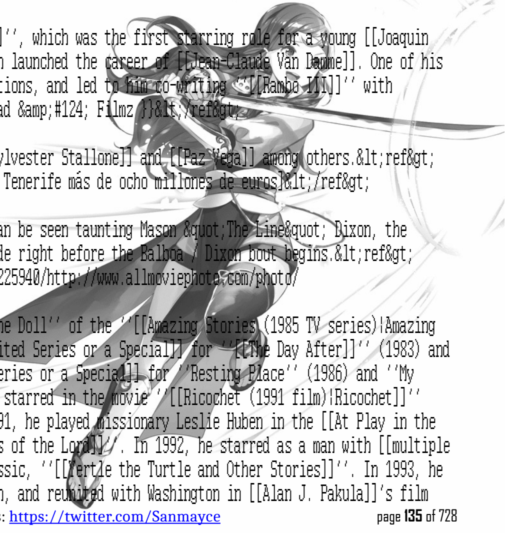
1578 |Co-wrote with [[Sylvester Stallone]]

1579 * In 2018, the surroundings of the National Park hosted part of the filming of the American film ''[[Rambo: Last Blood]]'', starring [[Sylvester Stallone]] and [[Paz Vega]] among others.&lt;ref&gt;[https://www.europapress.es/islas-canarias/noticia-rambo-dejara-tenerife-mas-ocho-millones-euros-20181113140555.html 'Rambo V' dejará en Tenerife más de ocho millones de euros]&lt;/ref&gt;

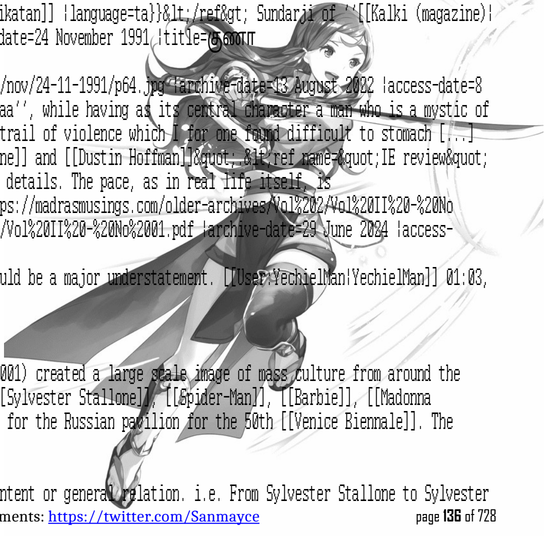
1580 | {{won}}{{efn|Tied with [[Sylvester Stallone]] for ''[[Rocky]]''.}}

1581 *''[[Rocky Balboa (film)|Rocky Balboa]]'' (2006), Tyson appears as himself in [[Sylvester Stallone]]'s 2006 movie ''Rocky Balboa''. He can be seen taunting Mason &quot;The Line&quot; Dixon, the current champion in the movie (who because of his arrogance and ability to end fights early, is based on Tyson's early career) at ringside right before the Balboa / Dixon bout begins.&lt;ref&gt;[http://www.allmoviephoto.com/photo/2006_rocky_balboa_032.html Rocky Balboa photo] {{webarchive|url=https://web.archive.org/web/20080403225940/http://www.allmoviephoto.com/photo/2006_rocky_balboa_032.html |date=April 3, 2008 }}, ''Allmoviephoto.com'', Retrieved on March 15, 2007.&lt;/ref&gt;

1582 In 1986, Lithgow received a [[Primetime Emmy Award for Outstanding Guest Actor in a Drama Series]] for his appearance in the episode ''The Doll'' of the ''[[Amazing Stories (1985 TV series)|Amazing Stories]]'' [[anthology series]]. Additionally, Lithgow has been nominated for an [[Emmy Award for Outstanding Supporting Actor in a Limited Series or a Special]] for ''[[The Day After]]'' (1983) and two [[Emmy Awards]] for [[Primetime Emmy Award for Outstanding Lead Actor in a Limited Series or Movie|Outstanding Lead Actor in a Miniseries or a Special]] for ''Resting Place'' (1986) and ''My Brother's Keeper'' (1995). Lithgow was approached about playing Dr. [[Frasier Crane]] on ''[[Cheers]]'', but turned it down. In 1991, he starred in the movie ''[[Ricochet (1991 film)|Ricochet]]'' opposite [[Denzel Washington]] as Earl Talbot Blake, a criminal seeking revenge against the policeman who sent him to prison. Also in 1991, he played missionary Leslie Huben in the [[At Play in the Fields of the Lord|film adaptation]] of [[Peter Matthiessen]]'s novel ''[[At Play in the Fields of the Lord (novel)|At Play in the Fields of the Lord]]''. In 1992, he starred as a man with [[multiple personality disorder]] in [[Brian De Palma]]'s film ''[[Raising Cain]]''. In 1992, he became the narrator in the [[Dr. Seuss]] video classic, ''[[Yertle the Turtle and Other Stories]]''. In 1993, he starred in [[Renny Harlin]]'s film ''[[Cliffhanger (film)|Cliffhanger]]'' opposite [[Sylvester Stallone]] as terrorist leader Eric Qualen, and reunited with Washington in [[Alan J. Pakula]]'s film



```
''[[The Pelican Brief (film)|The Pelican Brief]]''. In 1994, Lithgow played [[Franklin Delano Roosevelt]] in the [[NBC]] miniseries ''[[World War II: When Lions Roared]]'' (also known as ''Then There Were Giants'') directed by [[Joseph Sargent]] starring alongside [[Michael Caine]] and [[Bob Hoskins]]. He provided narration for the [[IMAX]] film ''[[Special Effects: Anything Can Happen]]'' (1996).
1583 * In December 2015, an auction brought in more than $3 million for memorabilia from [[Sylvester Stallone]]'s personal collection.&lt;ref&gt;{{Cite news|date=December 22, 2015|title=Sylvester Stallone's film memorabilia fetches $3 million at auction|work=Chicago Tribune|agency=Associated Press|url=https://www.chicagotribune.com/entertainment/ct-sylvester-stallone-film-memorabilia-auction-20151222-story.html|access-date=February 20, 2020}}&lt;/ref&gt;
1584 {{about|the 1981 Bollywood film|the Sylvester Stallone film|Rocky}}
1585 '''''Capone''''' is a 1975 American [[action film|action]] [[crime film]] directed by [[Steve Carver]], written by [[Howard Browne]], and starring [[Ben Gazzara]], [[Harry Guardino]], [[Susan Blakely]], [[John Cassavetes]], and [[Sylvester Stallone]] in an early film appearance. The film is a biography of the infamous gangster [[Al Capone]].&lt;ref&gt;{{cite book|page=78|title= Mind warp! : the fantastic true story of Roger Corman's New World Pictures|last=Koetting|first= Christopher T.|year=2013}}&lt;/ref&gt;
1586 * [[Sylvester Stallone]] as [[Frank Nitti|Frank Ralph &quot;The Enforcer&quot; Nitti]]
1587 Sylvester Stallone was cast on the basis of his appearance in ''Lords of Flatbush''. He later said &quot;I particularly enjoyed working on ''Capone'', because it was like the cheesy, mentally challenged inbred cousin of ''[[The Godfather]]''&quot;.&lt;ref&gt;[http://www.aintitcool.com/node/30932 Ain't It Cool News]&lt;/ref&gt; Stallone later made ''[[Death Race 2000]]'' for Corman.
1588 The film was released on [[DVD]] in the United States for the first time on March 29, 2011 through [[Shout! Factory]]&lt;ref&gt;{{cite web|url=https://www.amazon.com/Capone-Ben-Gazzara/dp/B004G7XCRQ |title=Capone: Ben Gazzara, Sylvester Stallone, John Cassavetes, Susan Blakely, Harry Guardino, Steve Carver: Movies &amp; TV |website=Amazon |date= |accessdate=2011-02-09}}&lt;/ref&gt; and has been available in Europe for some time.&lt;ref&gt;{{cite web|url=https://www.amazon.fr/Capone-Ben-Gazzara/dp/B000F7MFL0 |title=Capone: Ben Gazzara, Sylvester Stallone, John Cassavetes, Susan Blakely, Harry Guardino, Steve Carver: Movies &amp; TV |publisher=Amazon.fr}}&lt;/ref&gt;
1589 | rowspan=&quot;2&quot; | [[Sylvester Stallone]]
1590 &quot;The Woman in You&quot; was the last track recorded for ''Staying Alive'', and the lead single from the soundtrack.  Director [[Sylvester Stallone]] used the Bee Gees songs in the movie more as background music rather than the prominent way ''Saturday Night Fever'' had featured them. The single received more airplay than the Bee Gees previous two singles, though not enough to reach a top 10 position. In interviews following the release of the film, the brothers expressed their displeasure at the way their songs were edited and revealed that their hearts were not in the music. By 1983, the Bee Gees were focusing their talents on solo projects and production of other artists, so it is not surprising that they were not all enthusiastic about the ''Staying Alive'' movie.
1591 | [[Sylvester Stallone]]
1592 In 2014, Glazer opened the Unbreakable Performance Center in [[West Hollywood]] with former Chicago Bears' all-pro linebacker, [[Brian Urlacher]], and U.S. Women's Volleyball Captain, [[Lindsey Berg]].&lt;ref&gt;{{cite web |url=https://www.tmz.com/2014/06/02/brian-urlacher-jay-glazer-opening-new-gym-unbreakable-performance/|title=Brian Urlacher -- Opens Hollywood Baller Gym ... For Badass Athletes Only!}}&lt;/ref&gt; The gym has become the home to many elite athletes, actors, musicians, and business people. Clientele have included [[Demi Lovato]], [[Odell Beckham Jr.]], [[Chuck Liddell]], [[Michael Strahan]], [[Wiz Khalifa]], [[Chris Pratt]], and [[Sylvester Stallone]].&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/news/unbreakable-performance-center-demi-lovato-nick-jonas-sylvester-stallone-work-1010824|title=What It's Like Working Out with Pop Stars, Actors and Athletes|website=[[The Hollywood Reporter]]|date=6 June 2017}}&lt;/ref&gt;
1593     <title>Sylvester Stallone filmography</title>
1594 {{Short description|List of films featuring Sylvester Stallone}}
1595 American actor and filmmaker [[Sylvester Stallone]] has appeared in over 81 films. This is a list of his acting roles as well as directing, screenwriting, producing credits.
1596 | style=&quot;text-align:center;&quot; | &lt;ref&gt;{{cite web|url=http://www.worldation.com/stories/sylvester-stallones-journey-fame/|title=Sylvester Stallone's journey to fame|date=9 October 2017|access-date=21 December 2018|archive-date=9 March 2019|archive-url=https://web.archive.org/web/20190309235105/http://www.worldation.com/stories/sylvester-stallones-journey-fame/|url-status=dead}}&lt;/ref&gt;
1597 | style=&quot;text-align:center;&quot; | &lt;ref&gt;{{cite book|title=Farewell My Lovely : Robert Mitchum, Charlotte Rampling, John Ireland, Sylvia Miles, Anthony Zerbe, Harry Dean Stanton, Jack O'Halloran, Joe Spinell, Sylvester Stallone, Kate Murtagh, John O'Leary, Walter McGinn, John A. Alonzo, Dick Richards, Elliott Kastner, George Pappas, Jerry Bick, Jerry Bruckheimer, David Zelag Goodman, Raymond Chandler: Movies &amp; TV |date= |asin=6305460175 }}&lt;/ref&gt;
1598 | style=&quot;text-align:center;&quot; | &lt;ref&gt;{{cite web|url=https://archive.org/details/youtube-5UgAyfLi4Dk|title=Taxi 3 - Scene With Sylvester Stallone|last=Starty|date=24 September 2017|via=Internet Archive}}&lt;/ref&gt;
1599 | style=&quot;text-align:center;&quot; | &lt;ref name=&quot;Creed, Creed II&quot;&gt;{{cite web|url=https://www.cinemablend.com/news/2495651/how-sylvester-stallone-changed-creed-i-and-ii-to-make-it-more-comfortable|title=How Sylvester Stallone Changed Creed I And II To Make It More 'Comfortable'|first=Jason|last=Ingolfsland|date=2020-05-05|publisher=CinemaBlend|access-date=2020-11-18}}&lt;/ref&gt;
1600 | ''The Making of Rocky vs. Drago by Sylvester Stallone''
1601 | Episode: &quot;Sylvester Stallone&quot;
1602 | Episode: &quot;Sylvester Stallone/Jamiroquai&quot;
1603 | style=&quot;text-align:center;&quot; | &lt;ref&gt;{{cite web|url=http://ultimateclassicrock.com/sylvester-stallone-winner-takes-it-all-sammy-hagar/|title=Sylvester Stallone in 'Winner Takes It All' by Sammy Hagar|first=Chad|last=Childers|website=Ultimate Classic Rock|date=October 19, 2012 }}&lt;/ref&gt;
1604 | Episode: &quot;Sylvester Stallone&quot;; uncredited
1605 | style=&quot;text-align:center;&quot; | &lt;ref&gt;{{cite web|url=https://www.henson.com/jimsredbook/2013/01/191979/|title=1/9/1979 - 'Sylvester Stallone (MS)'|date=9 January 2013|access-date=18 December 2018|archive-date=15 May 2022|archive-url=https://web.archive.org/web/20220515044938/https://www.henson.com/jimsredbook/2013/01/191979/|url-status=dead}}&lt;/ref&gt;
1606 * [[List of awards and nominations received by Sylvester Stallone]]
1607 * {{IMDb name|230|Sylvester Stallone}}
1608 {{Sylvester Stallone}}
1609 [[Category:Sylvester Stallone|Filmography]]
1610 [[Category:Images of Sylvester Stallone]]</text>
1611 * [[Sylvester Stallone]]
1612 (Action figure versions of [[Arnold Schwarzenegger]] and [[Sylvester Stallone]] appear in this episode, though they do not voice themselves.)
1613 | data-sort-value=&quot;stallone&quot; |[[Sylvester Stallone]]
1614 He began his career at Warner Brothers, working for director [[Richard Donner]]&lt;ref name=&quot;Time Out&quot; /&gt; whose credits include ''[[Superman (1978 film)|Superman]]'', ''[[The Omen]]'', ''[[Scrooged]]'' and the ''[[Lethal Weapon (film series)|Lethal Weapon]]'' series.&lt;ref name = &quot;IMDB DD&quot;&gt;{{cite web|title=Richard Donner|url=https://www.imdb.com/name/nm0001149/|work=IMDB|accessdate=24 Aug 2016}}&lt;/ref&gt;  Goodloe worked as Donner's assistant from 1992 to 1995 on films such as ''[[Lethal Weapon 3]]'' and ''[[Maverick (film)|Maverick]]''.&lt;ref name = &quot;IMDB J Mills Goodloe&quot; /&gt;  In 1996, Goodloe was promoted to president of Donner Productions. Over the next four years, Goodloe produced ''[[Assassins (1995 film)|Assassins]]'', starring [[Sylvester Stallone]] and [[Julianne Moore]], ''[[Conspiracy Theory]]'' starring [[Mel Gibson]] and [[Julia Roberts]], and the final installment of the ''Lethal Weapon'' series starring Gibson, [[Danny Glover]] and [[Chris Rock]].&lt;ref name = &quot;IMDB J Mills Goodloe&quot; /&gt;  In 2001, Goodloe left Donner Productions to write and direct ''[[A Gentleman's Game]]'' starring [[Gary Sinise]], [[Dylan Baker]] and [[Philip Baker Hall]].&lt;ref name = &quot;IMDB J Mills Goodloe&quot; /&gt;   The film was nominated for an ESPY Award in the category of Best Sports Film.&lt;ref&gt;{{cite web |url=http://www.sportspectator.com/fancentral/lists/bestsportsmovies/espy.html |title=ESPY Awards Best Sports Movie Nominees and Winners |last=Jones|first=Bryan |publisher=Open Publishing |date=30 April 2005 |website=Sports Spectator |access-date=23 August 2016}}&lt;/ref&gt;
1615 On 24 November 1991, ''[[Ananda Vikatan]]'' appreciated the film and mentioned that Haasan had expanded the boundaries of his acting and had achieved a landmark success in this film, concluding that only Haasan could play this role effortlessly.&lt;ref&gt;{{Cite magazine |date=24 November 1991 |title=குணா |magazine=[[Ananda Vikatan]] |language=ta}}&lt;/ref&gt; Sundarji of ''[[Kalki (magazine)|Kalki]]'' appreciated the film for its cinematography, music and the cast performances.&lt;ref&gt;{{Cite magazine |last=சுந்தர்ஜி |date=24 November 1991 |title=குணா |url=https://kalkionline.com/kalkionline_archive/imagegallery/archiveimages/kalki/1991/nov/24-11-1991/p64.jpg |url-status=dead |archive-url=https://archive.today/20220813092258/https://kalkionline.com/kalkionline_archive/imagegallery/archiveimages/kalki/1991/nov/24-11-1991/p64.jpg |archive-date=13 August 2022 |access-date=8 November 2021 |magazine=[[Kalki (magazine)|Kalki]] |page=64 |language=Ta}}&lt;/ref&gt; ''[[The Indian Express]]'' wrote &quot;''Gunaa'', while having as its central character a man who is a mystic of sorts, a dreamer who wants to rise about the mire that he finds himself in and those around him hopelessly lodged in, lets loose a trail of violence which I for one found difficult to stomach [...] One comes out with the impression that the [Kamal Haasan] of the film is such a curious blend of [[Raj Kapoor]], [[Sylvester Stallone]] and [[Dustin Hoffman]]&quot;.&lt;ref name=&quot;IE review&quot; /&gt; Nalini Rajan of ''[[Madras Musings]]'' wrote, &quot;The film's genius really lies in the superfluous and seemingly irrelevant details. The pace, as in real life itself, is relentless&quot;.&lt;ref&gt;{{Cite news |last=Rajan |first=Nalini |date=16-30 April 1992 |title=The flip side's attractive |url=https://madrasmusings.com/older-archives/Vol%202/Vol%20II%20-%20No%2001.pdf |url-status=live |archive-url=https://web.archive.org/web/20240629114725/https://madrasmusings.com/older-archives/Vol%202/Vol%20II%20-%20No%2001.pdf |archive-date=29 June 2024 |access-date=5 July 2024 |work=[[Madras Musings]] |pages=4}}&lt;/ref&gt;
1616 *'''Delete''' and don't waste our time.  Nobody cares about Sylvester Stallone's three-year-old son.  To say he fails [[WP:BIO]] would be a major understatement. [[User:YechielMan|YechielMan]] 01:03, 26 April 2007 (UTC)
1617 [[File:Sylvester Stallone 2012.jpg|thumb|upright=0.8|[[Sylvester Stallone]]]]
1618 |[[Sylvester Stallone]]
1619 In 2001, Dubosarsky and Vinogradov made a transition from the socialist fantasy to the ideals of [[mass media]].  Total Painting (2001) created a large scale image of mass culture from around the world borrowing ideas from [[advertising]] and mainstream media.  Our Best World (2003) created a common ground for stars such as [[Sylvester Stallone]], [[Spider-Man]], [[Barbie]], [[Madonna (entertainer)|Madonna]], [[Arnold Schwarzenegger]], [[Marilyn Monroe]], and even [[Picasso]].  The Underwater World (2002) was made for the Russian pavilion for the 50th [[Venice Biennale]]. The piece references ideas gathered from pop images such as mail order catalogs, and heroes in contemporary advertising.
1620 * [[Jackie Stallone]] (1921-2020), American astrologer, mother of actor Sylvester Stallone and singer Frank Stallone
1621 On a web forum we play a game of 'link the photo'. This consists of, as the title implies, linking one photo to another by title,content or general relation. i.e. From Sylvester Stallone to Sylvester
```



the cat to Tweetie Pie and so on.

1622 |{{sort|Stallone|[[File:Sylvester Stallone 2012.jpg|50px]]}}

1623 ''Popsim'' was published by [[Harcourt Brace Jovanovich]] in March 1980. [[Bob Colacello]] hosted a dinner party for the launch of the book at La Boite in New York City on Mach 24, 1980.&lt;ref&gt;{{Cite book |last=Warhol |first=Andy |url= |title=The Andy Warhol Diaries |last2=Hackett |first2=Pat |date=1989 |publisher=Warner Books |others= |isbn=978-0-446-51426-2 |location=New York, NY |pages=274 |postscript=Diary Entry: March 24, 1980}}&lt;/ref&gt; The guests included [[Henry Geldzahler]], [[Ahmet Ertegun]], [[Richard Gere]], [[Sylvester Stallone]], [[Bianca Jagger]], [[Debbie Harry]], and [[Paloma Picasso]].&lt;ref&gt;{{Cite news |last=Suzy |date=1980-03-26 |title=Astaire's 'Fine Romance' jars family |url=https://www.newspapers.com/article/daily-news/151979368/ |access-date=2024-07-25 |work=Daily News |pages=340}}&lt;/ref&gt;

1624 Courtleigh got his start in the 1950s science fiction television series ''[[Atom Squad]]'' playing character Steve Elliott alongside [[Bob Hastings]] and [[Bram Nossem]]. This role led to Courtleigh being chosen as an honorary delegate to the ninth annual [[World Science Fiction Convention]] held in Philadelphia, Pennsylvania.&lt;ref&gt;{{cite web |title=Atom Squad Came Out of WPTZ, Channel 3 in Philadelphia |url=http://broadcastpioneers.50megs.com/atomsquad.html |website=Broadcast Pioneers of Philadelphia |archiveurl=https://web.archive.org/web/20040617024406/http://broadcastpioneers.50megs.com/atomsquad.html |archivedate=17 June 2004}}&lt;/ref&gt; Among Courtleigh's other television roles were an episode of ''[[Men into Space]]'' in the 1960s, the part of the evil stepmother's groom in [[Rodgers and Hammerstein]]'s ''[[Cinderella (Rodgers and Hammerstein musical)|Cinderella]]'', a [[CBS]] TV special starring [[Lesley Ann Warren]]; several appearances in [[Hallmark Hall of Fame]] productions; and supporting roles in the [[Sylvester Stallone]] drama ''[[F.I.S.T. (film)|F.I.S.T.]]'' and ''[[Winter Kills (film)|Winter Kills]]'', a [[John Huston]] thriller.

1625 | ''[[Rambo: First Blood Part II]]'' || [[TriStar Pictures|Tri-Star Pictures]] || [[George P. Cosmatos]] (director); [[Sylvester Stallone]], [[James Cameron]] (screenplay); [[Sylvester Stallone]], [[Richard Crenna]], [[Charles Napier (actor)|Charles Napier]], [[Steven Berkoff]], [[Julia Nickson]], [[Martin Kove]], [[George Cheung]], Andy Wood, William Ghent, Voyo Goric, Dana Lee, Steve Williams

1626 | ''[[Rocky IV]]'' || [[Metro-Goldwyn-Mayer|MGM/UA Entertainment Company]] || [[Sylvester Stallone]] (director/screenplay); [[Sylvester Stallone]], [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], [[Dolph Lundgren]], [[Brigitte Nielsen]], [[Tony Burton]], [[Michael Pataki]], Rocky Krakoff, [[LeRoy Neiman]], [[Burgess Meredith]], [[James Brown]], [[Stu Nahan]], [[Warner Wolf]], [[R.J. Adams]], [[Barry Tompkins]], [[Al Bandiero]], David Lloyd Austin

1627 | ''[[Lock Up (1989 film)|Lock Up]]'' || [[TriStar Pictures|Tri-Star Pictures]] / [[Carolco Pictures]] || [[John Flynn (director)|John Flynn]] (director); Richard Smith, [[Jeb Stuart (writer)|Jeb Stuart]], Henry Rosenbaum (screenplay); [[Sylvester Stallone]], [[Donald Sutherland]], [[John Amos]], [[Darlanne Fluegel]], [[Frank McRae]], [[Sonny Landham]], [[Tom Sizemore]], [[William Allen Young]], [[Larry Romano]], [[Jordan Lund]], [[Danny Trejo]], [[Frank Pesce]], [[Anthony Crivello]], John Lilla, Dean Duval, Jerry Strivelli, David Anthony Marshall, Kurek Ashley, Michael Petroni

1628 | ''[[Tango &amp; Cash]]'' || [[Warner Bros. Pictures]] || [[Andrei Konchalovsky]] (director); Randy Feldman (screenplay); [[Sylvester Stallone]], [[Kurt Russell]], [[Jack Palance]], [[Teri Hatcher]], [[Brion James]], [[Geoffrey Lewis (actor)|Geoffrey Lewis]], [[Eddie Bunker]], [[James Hong]], [[Marc Alaimo]], [[Michael J. Pollard]], [[Michael Jeter]], [[Robert Z'Dar]], [[Lewis Arquette]], [[Roy Brocksmith]], [[Richard Fancy]], [[Phil Rubenstein]], [[Clint Howard]], [[Benny Urquidez]], [[Billy Blanks]], [[Kristen Dalton (actress)|Kristen Dalton]], [[Dale Swann]], [[Glenn Morshower]], [[Bing Russell]], [[Elizabeth Sung]], [[Andre Rosey Brown]], [[Savely Kramarov]], [[Patti Davis]], [[Roxanne Kernohan]], [[Adolfo Quinones|Adolfo 'Shabba-Doo' Quinones]], [[Rodney Saulsberry]]

1629 | ''[[Over the Top (1987 film)|Over the Top]]'' || [[Warner Bros. Pictures]] / [[Cannon Films]] || [[Menahem Golan]] (director); [[Stirling Silliphant]], [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[Robert Loggia]], [[Susan Blakely]], [[David Mendenhall]], [[Rick Zumwalt]], [[Terry Funk]], [[Jimmy Keegan]], [[Allan Graf]], [[Reggie Bennett]], [[Scott Norton]], [[John Brzenk]], Chris McCarty, Bruce Way, Greg &quot;Magic&quot; Schwartz, John Braden, Allen Fisher, John Vreeland, Cleve Dean, Andrew &quot;Cobra&quot; Rhodes

1630 | ''[[The Prisoner of Second Avenue]]'' || [[Warner Bros.]] || [[Melvin Frank]] (director); [[Neil Simon]] (screenplay); [[Jack Lemmon]], [[Anne Bancroft]], [[Gene Saks]], [[Elizabeth Wilson]], [[Florence Stanley]], [[Maxine Stuart]], [[Ed Peck]], [[Ivor Francis]], [[Stack Pierce]], [[Patricia Marshall]], [[Ketty Lester]], [[M. Emmet Walsh]], [[Sylvester Stallone]], [[F. Murray Abraham]], [[Lonnie Burr]], [[Gary Owens]], [[John Ritter]], [[Joe Turkel]], Gene Blakely, Dee Carroll, James McCallion, Alan DeWitt, Cosmo Sardo, Norman Stevans

1631 | ''[[Capone (1975 film)|Capone]]'' || [[20th Century Fox]] / Santa Fe Productions || [[Steve Carver]] (director); [[Howard Browne]] (screenplay); [[Ben Gazzara]], [[Harry Guardino]], [[Susan Blakely]], [[John Cassavetes]], [[Sylvester Stallone]], [[Frank Campanella]], [[John Orchard]], [[Carmen Argenziano]], [[George Chandler]], [[John Davis Chandler]], [[Royal Dano]], [[Peter Maloney (actor)|Peter Maloney]], [[Dick Miller]], [[Robert Phillips (actor)|Robert Phillips]], [[Martin Kove]], [[Mario Gallo (actor)|Mario Gallo]], [[Tony Giorgio (actor)|Tony Giorgio]], [[Johnny Martino]], [[Tina Scala]]

1632 | ''[[Death Race 2000]]'' || [[New World Pictures]] || [[Paul Bartel]] (director); [[Robert Thom (writer)|Robert Thom]], [[Charles B. Griffith]] (screenplay); [[Sylvester Stallone]], [[David Carradine]], [[Simone Griffeth]], [[Louisa Moritz]], [[Don Steele]], [[Mary Woronov]], [[Roberta Collins]], [[Martin Kove]], [[Joyce Jameson]], [[Paul Laurence]], [[Harriet Medin]], [[Bill Morey]], [[Fred Grandy]], [[John Landis]], Carle Bensen, Sandy McCallum, Vince Trankina, William Shephard, Leslie McRay, Wendy Bartel, Jack Favorite, Sandy Ignon, Darla McDonell, Roger Rook

1633 | ''[[Mandingo (film)|Mandingo]]'' || [[Paramount Pictures]] / [[Dino De Laurentiis Company]] || [[Richard Fleischer]] (director); [[Norman Wexler]] (screenplay); [[James Mason]], [[Susan George (actress)|Susan George]], [[Perry King]], [[Richard Ward (actor)|Richard Ward]], [[Brenda Sykes]], [[Ken Norton]], [[Lillian Hayman]], [[Sylvester Stallone]]

1634 | ''[[Farewell, My Lovely (1975 film)|Farewell, My Lovely]]'' || [[AVCO Embassy Pictures]] / [[ITC Entertainment]] || [[Dick Richards]] (director); [[David Zelag Goodman]] (screenplay); [[Robert Mitchum]], [[Charlotte Rampling]], [[John Ireland (actor)|John Ireland]], [[Sylvia Miles]], [[Anthony Zerbe]], [[Harry Dean Stanton]], [[Jack O'Halloran]], [[Joe Spinell]], [[Sylvester Stallone]], [[Rainbeaux Smith]], [[Kate Murtagh]], John O'Leary, [[Walter McGinn]], [[Burton Gilliam]], [[Jim Thompson (writer)|Jim Thompson]], Jimmie Archer, [[Ted Gehring]]

1635 | ''[[Rocky II]]'' || [[United Artists]] / Chartoff-Winkler Productions || [[Sylvester Stallone]] (director/screenplay); [[Sylvester Stallone]], [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], [[Burgess Meredith]], [[Tony Burton]], Sylvia Meals, Seargeoh Stallone, [[Joe Spinell]], Paul J. Micale, Jeff Temkin, [[Lou Filippo]], [[Brent Musburger]], [[Stu Nahan]], Bill Baldwin, [[LeRoy Neiman]]

1636 | ''[[Rambo III]]'' || [[TriStar Pictures|Tri-Star Pictures]] / [[Carolco Pictures]] || [[Peter MacDonald (director)|Peter MacDonald]] (director); [[Sylvester Stallone]], [[Sheldon Lettich]] (screenplay); [[Sylvester Stallone]], [[Richard Crenna]], [[Kurtwood Smith]], [[Marc de Jonge]], [[Sasson Gabai]], Doudi Shoua, [[Spiros Focas]], Randy Raney, [[Marcus Gilbert (actor)|Marcus Gilbert]], [[Alon Abutbul]], Mahmoud Assadollahi, [[Yosef Shiloach|Yosef Shiloah]], Shaby Ben-Aroya

1637 | ''[[Rocky V]]'' || [[United Artists]] || [[John G. Avildsen]] (director); [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[Talia Shire]], [[Burt Young]], [[Sage Stallone]], [[Richard Gant]], [[Tommy Morrison]], [[Burgess Meredith]], [[Tony Burton]], Paul J. Micale, [[Michael Williams (boxer)|Michael Williams]], [[Kevin Connolly (actor)|Kevin Connolly]], [[Al Bernstein]], [[Stu Nahan]]

1638 | ''[[Oscar (1991 film)|Oscar]]'' || [[Touchstone Pictures]] || [[John Landis]] (director); Michael Barrie, [[Jim Mulholland]] (screenplay); [[Sylvester Stallone]], [[Marisa Tomei]], [[Ornella Muti]], [[Don Ameche]], [[Peter Riegert]], [[Tim Curry]], [[Vincent Spano]], [[Eddie Bracken]], [[Linda Gray]], [[Chazz Palminteri]], [[Kurtwood Smith]], [[Art LaFleur]], [[Robert Lesser]], [[Yvonne De Carlo]], [[Martin Ferrero]], [[Harry Shearer]], [[Richard Romanus]], [[Arleen Sorkin]], [[Joey Travolta]], [[Jim Mulholland]], [[Kirk Douglas]], Elizabeth Barondes, [[Joycelyn O'Brien]]

1639 | ''[[Stop! Or My Mom Will Shoot]]'' || [[Universal Pictures]] / &lt;small&gt;[[Ivan Reitman|Northern Lights Entertainment]]&lt;/small&gt; || [[Roger Spottiswoode]] (director); [[Blake Snyder]], [[William Osborne (writer)|William Osborne]], [[William Davies (screenwriter)|William Davies]] (screenplay); [[Sylvester Stallone]], [[Estelle Getty]], [[JoBeth Williams]], Al Fann, [[Roger Rees]], [[Martin Ferrero]], [[Gailard Sartain]], [[John Wesley (actor)|John Wesley]], [[J. Kenneth Campbell]], [[Ving Rhames]], [[Richard Schiff]], [[Dennis Burkley]], [[Ella Joyce]], [[Nicholas Sadler]], [[Vanessa Angel (English actress)|Vanessa Angel]], [[Marjean Holden]], [[Patti Yasutake]], [[Jophery Brown]], [[Manny Perry]], [[Ernie Lively]], [[Julia Montgomery]], [[Dorian Gregory]]

1640 | ''[[Men in Black (1997 film)|Men in Black]]'' || [[Columbia Pictures]] / &lt;small&gt;[[Amblin Entertainment]]&lt;/small&gt; || [[Barry Sonnenfeld]] (director); [[Ed Solomon]] (screenplay); [[Tommy Lee Jones]], [[Will Smith]], [[Linda Fiorentino]], [[Vincent D'Onofrio]], [[Rip Torn]], [[Tony Shalhoub]], [[Siobhan Fallon Hogan|Siobhan Fallon]], [[Mike Nussbaum]], [[Jon Gries]], Sergio Calderon, [[Carel Struycken]], [[Fredric Lehne]], [[Richard Hamilton (actor)|Richard Hamilton]], [[Kent Faulcon]], John Alexander, Keith Campbell, [[Ken Thorley]], [[Patrick Breen]], [[Becky Ann Baker]], [[Sean Whalen]], [[Michael Willis]], [[Willie C. Carpenter]], [[David Cross]], [[Debbie Lee Carrington]], [[Verne Troyer]], [[Tim Blaney]], [[Harsh Nayyar]], Mark Setrakian, Brad Abrell, Thom Fountain, Carl J. Johnson, [[Drew Massey]], [[Lowell Cunningham]], [[Danny DeVito]], [[Newt Gingrich]], [[Karen Lynn Gorney]], [[George Lucas]], [[Isaac Mizrahi]], [[Tony Robbins]], [[Al Roker]], [[Barry Sonnenfeld]], [[Steven Spielberg]], [[Sylvester Stallone]], [[Dionne Warwick]]

1641 | ''[[Cop Land]]'' || [[Miramax|Miramax Films]] || [[James Mangold]] (director/screenplay); [[Sylvester Stallone]], [[Harvey Keitel]], [[Ray Liotta]], [[Robert De Niro]], [[Peter Berg]], [[Janeane Garofalo]], [[Robert Patrick]], [[Michael Rapaport]], [[Annabella Sciorra]], [[Cathy Moriarty]], [[Noah Emmerich]], [[John Spencer (actor)|John Spencer]], [[Frank Vincent]], [[Malik Yoba]], [[Arthur Nascarella]], [[Victor Williams]], [[Edie Falco]], [[Mel Gorham]], [[Paul Calderón]], [[Vincent Laresca]], [[Method Man]], [[Deborah Harry]], [[Tony Sirico]]

1642 | ''[[Cliffhanger (film)|Cliffhanger]]'' || [[TriStar Pictures]] / [[Carolco Pictures]] || [[Renny Harlin]] (director); [[Michael France]], [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[John Lithgow]], [[Michael Rooker]], [[Janine Turner]], [[Rex Linn]], [[Caroline Goodall]], [[Leon Robinson|Leon]], [[Craig Fairbrass]], [[Gregory Scott Cummins]], [[Denis Forest]], Michelle Joyner, [[Paul Winfield]], [[Ralph Waite]], [[Max Perlich]], Trey Brownell, [[Zach Grenier]], [[Vyto Ruginis]], [[Don S. Davis]], Scott Hoxby, [[John Finn]], [[Bruce McGill]], [[Rosemary Dunsmore]], [[Jeff McCarthy]]

1643 | ''[[Demolition Man (film)|Demolition Man]]'' || [[Warner Bros. Pictures]] / &lt;small&gt;[[Silver Pictures]]&lt;/small&gt; || [[Marco Brambilla]] (director); [[Daniel Waters (screenwriter)|Daniel Waters]], Robert Reneau, Peter M. Lenkov (screenplay); [[Sylvester Stallone]], [[Wesley Snipes]], [[Sandra Bullock]], [[Nigel Hawthorne]], [[Benjamin Bratt]], [[Denis Leary]], [[Bill Cobbs]], [[Bob Gunton]], [[Glenn Shadix]], [[Trent Walker]], [[Troy Evans (actor)|Troy Evans]], [[David Patrick Kelly]], [[Steve Kahan]], [[Andre Gregory]], [[Jack Black]], [[Jesse Ventura]], [[Brandy Ledford]], [[Grand L. Bush]], [[Mark Colson]]

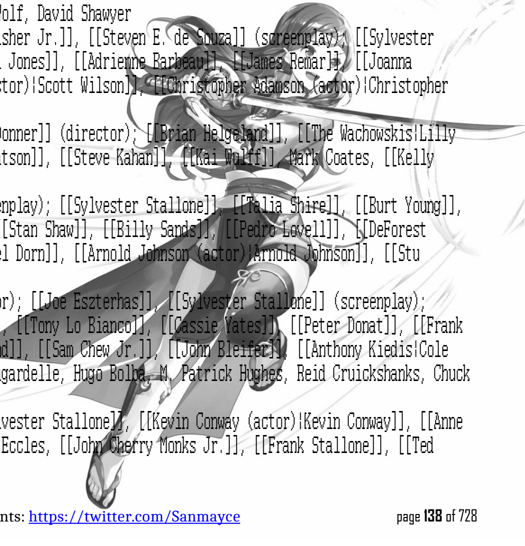
1644 | ''[[The Lords of Flatbush]]'' || [[Columbia Pictures]] || [[Martin Davidson]], [[Stephen Verona|Stephen F. Verona]] (directors/screenplay); Gayle Gleckler, [[Sylvester Stallone]] (screenplay); [[Perry King]], [[Sylvester Stallone]], [[Henry Winkler]], [[Susan Blakely]], [[Paul Jabara]], [[Martin Davidson]], [[Joseph Stern]], [[Dolph Sweet]], [[Antonia Rey]], [[Ray Sharkey]], [[Geraldine Smith (actress)|Geraldine Smith]]\

1645 | ''[[The Specialist]]'' || [[Warner Bros. Pictures]] || [[Luis Llosa]] (director); Alexandra Seros (screenplay); [[Sylvester Stallone]], [[Sharon Stone]], [[James Woods]], [[Eric Roberts]], [[Rod Steiger]], [[Mario Ernesto Sanchez]], Chase Randolph, Jeana Bell, Tony Munafo, [[Emilio Estefan, Jr.]]

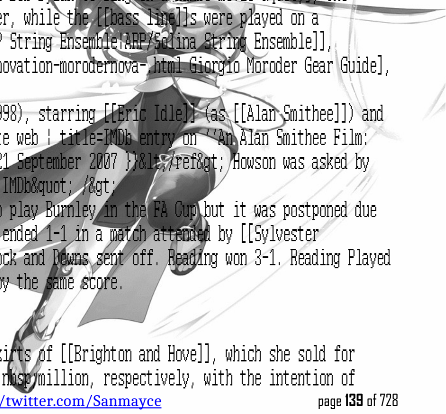
1646 | ''[[Get Carter (2000 film)|Get Carter]]'' || [[Warner Bros. Pictures]] / [[Morgan Creek Entertainment|Morgan Creek Productions]] / [[Franchise Pictures]] / [[Mark Canton|The Canton Company]] || [[Stephen Kay]] (director); [[David McKenna (writer)|David McKenna]] (screenplay); [[Sylvester Stallone]], [[Miranda Richardson]], [[Rachael Leigh Cook]], [[Alan Cumming]], [[Mickey Rourke]], [[John C. McGinley]], [[Michael Caine]], [[Rhona Mitra]], [[Johnny Strong]], [[John Cassini]], [[Mark Boone Junior]], [[Gretchen Mol]], [[Tom Sizemore]], [[Crystal Lowe]]

1647 | ''[[Driven (2001 film)|Driven]]'' || [[Warner Bros. Pictures]] / [[Franchise Pictures]] || [[Renny Harlin]] (director); [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[Burt Reynolds]], [[Kip Pardue]], [[Til Schweiger]], [[Gina Gershon]], [[Estella Warren]], [[Cristián de la Fuente]], [[Stacy Edwards]], [[Robert Sean Leonard]], [[Jean Alesi]], [[Jacques Villeneuve]],

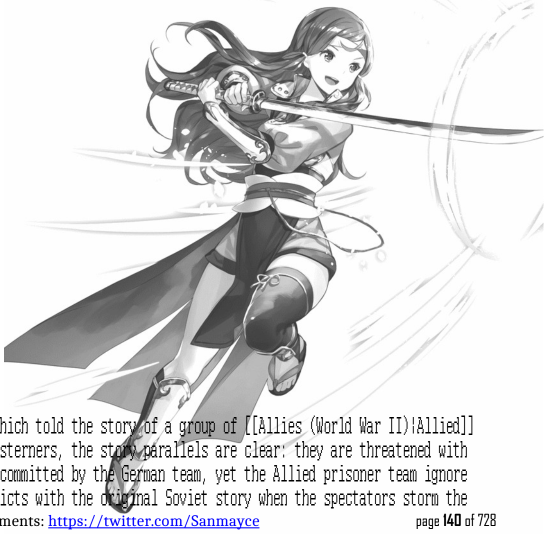
```
[[Juan Pablo Montoya]], [[Adrián Fernández]], [[Mark Blundell]], [[Roberto Moreno]], [[Kenny Brack]], [[Tony Kanaan]], [[Mauricio Gugelmin]], [[Max Papis]]
¦ ''[[Rocky III]]'' ¦¦ [[MGM/UA Entertainment Co.]] / [[United Artists]] ¦¦ [[Sylvester Stallone]] (director/screenplay); [[Sylvester Stallone]], [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], [[Burgess Meredith]], [[Mr. T]], [[Tony Burton]], [[Ian Fried (actor)¦Ian Fried]], [[Hulk Hogan]], [[Stu Nahan]], [[Jimmy Lennon]], [[Lou Filippo]], [[Dennis James]], [[Jim Healy (sports commentator)¦Jim Healy]], [[LeRoy Neiman]], [[Jim Hill (American football)¦Jim Hill]], [[Morgan Freeman]], [[Jim Henson]], Bill Baldwin, Marty Denkin
¦ ''[[First Blood]]'' ¦¦ [[Orion Pictures]] / Anabasis Investments, N.V. ¦¦ [[Ted Kotcheff]] (director); [[Michael Kozoll]], William Sackheim, [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[Richard Crenna]], [[Brian Dennehy]], [[Bill McKinney]], [[Jack Starrett]], [[Michael Talbott]], [[Chris Mulkey]], [[John McLiam]], [[Alf Humphreys]], [[David Caruso]], David L. Crowley, Don MacKay, Patrick Stark
¦ ''[[Spy Kids 3-D: Game Over]]'' ¦¦ [[Dimension Films]] / [[Troublemaker Studios]] ¦¦ [[Robert Rodriguez]] (director/screenplay); [[Antonio Banderas]], [[Carla Gugino]], [[Alexa PenaVega¦Alexa Vega]], [[Daryl Sabara]], [[Ricardo Montalbán]], [[Holland Taylor]], [[Sylvester Stallone]], [[Mike Judge]], [[Salma Hayek]], [[Matt O'Leary]], [[Emily Osment]], [[Ryan Pinkston]], Robert Vito, [[Bobby Edner]], [[Courtney Jines]], [[Cheech Marin]], [[Danny Trejo]], [[Alan Cumming]], [[Tony Shalhoub]], [[Steve Buscemi]], [[Bill Paxton]], [[George Clooney]], [[Elijah Wood]], [[Selena Gomez]], [[Glen Powell]], [[James Paxton (actor)¦James Paxton]] ¦¦
¦ ''[[Avenging Angelo]]'' ¦¦ [[Warner Bros. Pictures]] / [[Franchise Pictures]] / [[DEJ Productions]] ¦¦ [[Martyn Burke]] (director); Will Aldis, [[Steve Mackall]] (screenplay); [[Sylvester Stallone]], [[Madeleine Stowe]], [[Anthony Quinn]], [[Raoul Bova]], [[Harry Van Gorkum]], [[Billy Gardell]], George Touliatos ¦¦
¦ ''[[Bananas (film)¦Bananas]]'' ¦¦ [[United Artists]] ¦¦ [[Woody Allen]] (director/screenplay); [[Woody Allen]], [[Louise Lasser]], [[Carlos Montalban]], [[Natividad Abascal]], [[Howard Cosell]], [[Jacobo Morales]], [[Miguel Ángel Suárez]], [[René Enríquez]], [[Roger Grimsby]], [[Don Dunphy]], [[Dan Frazer]], [[Charlotte Rae]], [[Axel Anderson]], [[Arthur Hughes (American actor)¦Arthur Hughes]], [[Eddie Barth]], [[Nicholas Saunders (actor)¦Nicholas Saunders]], [[Conrad Bain]], [[Allen Garfield]], [[Hy Anzell]], [[Sylvester Stallone]], [[Mary Jo Catlett]], [[Tino García]], David Ortiz, Jack Axelrod, [[Martha Greenhouse]], Stanley Ackerman
¦ ''[[Klute]]'' ¦¦ [[Warner Bros.]] / Gus Productions ¦¦ [[Alan J. Pakula]] (director); [[Andy Lewis (screenwriter)¦Andy Lewis]], Dave Lewis (screenplay); [[Jane Fonda]], [[Donald Sutherland]], [[Charles Cioffi]], [[Roy Scheider]], Dorothy Tristan, [[Rita Gam]], [[Vivian Nathan]], [[Nathan George]], [[Jane White]], [[Shirley Stoler]], [[Robert Milli]], [[Anthony Holland (actor)¦Anthony Holland]], [[Richard B. Shull]], [[Mary Louise Wilson]], [[Rosalind Cash]], [[Jean Stapleton]], [[Antonia Rey]], [[Candy Darling]], [[Kevin Dobson]], [[Veronica Hamel]], [[Richard Jordan]], [[Harry Reems]], [[Joe Silver]], [[Lee Wallace (actor)¦Lee Wallace]], [[Sylvester Stallone]]
*&quot;Rum casusu&quot; features a sample of [[Ferruccio Amendola]] dubbing [[Sylvester Stallone]] in ''[[Rocky V]]'', telling a joke based on the lyrics of Elio e le Storie Tese's 1989 song &quot;Nubi di ieri sul nostro domani odierno (Abitudinario)&quot;.&lt;ref name=marok/&gt;
¦ ''[[Rocky Balboa (film)¦Rocky Balboa]]'' ¦¦ [[Metro-Goldwyn-Mayer]] / [[Columbia Pictures]] / [[Revolution Studios]] ¦¦ [[Sylvester Stallone]] (director/screenplay); [[Sylvester Stallone]], [[Burt Young]], [[Antonio Tarver]], [[Milo Ventimiglia]], [[Geraldine Hughes]], [[Tony Burton]], [[Lou DiBella]], James Francis Kelly III, [[Pedro Lovell]], [[Jacob Duran¦Jacob &quot;Stitch&quot; Duran]], [[Talia Shire]], [[A.J. Benza]], [[Mike Tyson]], [[Henry G. Sanders]], [[Woody Paige]], [[Skip Bayless]], [[Jay Crawford]], [[Brian Kenny (sportscaster)¦Brian Kenny]], [[Dana Jacobson]], [[Jim Lampley]], [[Larry Merchant]], [[Max Kellerman]], [[LeRoy Neiman]], [[Bert Sugar]], [[Michael Buffer]], [[Joe Cortez]], [[Frank Stallone]], [[Tobias Segal]], [[Stu Nahan]], [[Dolph Lundgren]], [[Burgess Meredith]], [[Mr. T]]
¦ ''[[Rambo (2008 film)¦Rambo]]'' ¦¦ [[Lionsgate Films¦Lionsgate]] / [[The Weinstein Company]] / [[Alchemy (company)¦Millennium Films]] / [[Avi Lerner¦Equity Pictures]] ¦¦ [[Sylvester Stallone]] (director/screenplay); Art Monterastelli (screenplay); [[Sylvester Stallone]], [[Julie Benz]], [[Paul Schulze]], [[Matthew Marsden]], [[Graham McTavish]], [[Tim Kang]], [[Rey Gallegos]], [[Jake La Botz]], [[Ken Howard]], [[Supakorn Kitsuwon]], Maung Maung Khin, Aung Aay Noi, Sornram Patchimtasanakarn
¦ ''[[The Party at Kitty and Stud's]]'' ¦¦ Cinema Epoch / Italian Stallion Productions ¦¦ Morton Lewis (director/screenplay); [[Sylvester Stallone]], Henrietta Holm, Jodi Van Prang, Nicholas Warren, Frank Micelli, Barbara Storm, [[Janet Banzet]]
¦ ''[[The Sidelong Glances of a Pigeon Kicker]]'' ¦¦ [[Metro-Goldwyn-Mayer]] / Saturn Productions ¦¦ [[John Dexter]] (director); [[Ron Whyte]] (screenplay); [[Jordan Christopher]], [[Jill O'Hara]], [[Robert Walden]], [[Kate Reid]], [[William Redfield (actor)¦William Redfield]], [[Lois Nettleton]], [[Elaine Stritch]], [[Melba Moore]], [[Kristoffer Tabori]], [[Sylvester Stallone]], Boni Enten, Riggs O'Hara, Donald Warfield, Jean Shevlin, Matt Warner
¦ ''[[An Alan Smithee Film: Burn Hollywood Burn]]'' ¦¦ [[Hollywood Pictures]] / [[Cinergi Pictures]] ¦¦ [[Arthur Hiller]] (director); [[Joe Eszterhas]] (screenplay); [[Eric Idle]], [[Ryan O'Neal]], [[Coolio]], [[Chuck D]], [[Richard Jeni]], [[Leslie Stefanson]], [[Sandra Bernhard]], [[Cherie Lunghi]], [[Harvey Weinstein]], [[Gavin Polone]], [[MC Lyte]], [[Marcello Thedford]], [[Stephen Tobolowsky]], [[Erik King]], [[Dina Spybey]], [[Sylvester Stallone]], [[Whoopi Goldberg]], [[Jackie Chan]], [[Robert Evans]], [[Robert Shapiro (film producer)¦Robert Shapiro]], [[Shane Black]], [[Mario Machado]], [[Lisa Canning]], [[Joe Eszterhas]], [[Larry King]], [[Peter Bart]], [[Dominick Dunne]], [[Billy Bob Thornton]], [[Billy Barty]], [[Nicole Nagel]], [[Jim Piddock]], [[Naomi Campbell]], [[Marianne Muellerleile]], [[Suli McCullough]], [[Duane Davis]], [[Stanley Ralph Ross]], [[Norman Jewison]]
¦ ''[[54 (film)¦54]]'' ¦¦ [[Miramax¦Miramax Films]] ¦¦ [[Mark Christopher (director)¦Mark Christopher]] (director/screenplay); [[Ryan Phillippe]], [[Salma Hayek]], [[Neve Campbell]], [[Mike Myers]], [[Sela Ward]], [[Breckin Meyer]], [[Sherry Stringfield]], [[Cameron Mathison]], [[Heather Matarazzo]], [[Skipp Sudduth]], [[Mark Ruffalo]], [[Lauren Hutton]], [[Michael York]], [[Ellen Albertini Dow]], [[Peter Bogdanovich]], [[Cindy Crawford]], [[Sheryl Crow]], [[Elio Fiorucci]], [[Bruce Jay Friedman]], [[Art Garfunkel]], [[Georgina Grenville]], [[Thelma Houston]], [[Ron Jeremy]], [[Beverly Johnson]], [[Heidi Klum]], [[Lorna Luft]], [[Valerie Perrine]], [[Stars on 54]], [[Cecilie Thomsen]], [[Donald Trump]], [[Frederique van der Wal]], [[Veronica Webb]], [[Sylvester Stallone]]
¦ ''[[Antz]]'' ¦¦ [[DreamWorks Pictures¦DreamWorks]] / &lt;small&gt;[[Pacific Data Images]]&lt;/small&gt; ¦¦ [[Eric Darnell]], [[Tim Johnson (film director)¦Tim Johnson]] (directors); [[Paul Weitz (filmmaker)¦Paul Weitz]], [[Chris Weitz]], [[Todd Alcott]] (screenplay); [[Woody Allen]], [[Gene Hackman]], [[Sharon Stone]], [[Sylvester Stallone]], [[Jennifer Lopez]], [[Christopher Walken]], [[Anne Bancroft]], [[Dan Aykroyd]], [[Danny Glover]], [[John Mahoney]], [[Jane Curtin]], [[Grant Shaud]], [[Paul Mazursky]], [[Jim Cummings]], [[April Winchell]], [[Jack Angel]], [[Rodger Bumpass]], [[Bill Farmer]], [[Patrick Pinney]], [[Phil Proctor]], Eric Darnell, Mickie McGowan, Gary Schwartz, Jerry Sroka, Carole Jeghers, Marty Sixkiller
¦ ''[[Cobra (1986 film)¦Cobra]]'' ¦¦ [[Warner Bros. Pictures]] / [[The Cannon Group, Inc.¦Cannon Films]] ¦¦ [[George P. Cosmatos]] (director); [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[Brigitte Nielsen]], [[Reni Santoni]], [[Andrew Robinson (actor)¦Andrew Robinson]], [[Brian Thompson]], [[John Herzfeld]], [[Lee Garlington]], [[Art LaFleur]], [[Marco Rodríguez (actor)¦Marco Rodríguez]], [[Val Avery]], [[David Rasche]], Nick Angotti, [[Harry Demopoulos]]
¦ ''[[Daylight (1996 film)¦Daylight]]'' ¦¦ [[Universal Pictures]] ¦¦ [[Rob Cohen]] (director); [[Leslie Bohem]] (screenplay); [[Sylvester Stallone]], [[Amy Brenneman]], [[Viggo Mortensen]], [[Dan Hedaya]], [[Jay O. Sanders]], [[Karen Young (actress)¦Karen Young]], [[Claire Bloom]], [[Danielle Harris]], [[Barry Newman]], [[Stan Shaw]], [[Colin Fox (actor)¦Colin Fox]], [[Vanessa Bell Calloway]], [[Sage Stallone]], [[Renoly Santiago]], [[Trina McGee]], [[Marcello Thedford]], [[Mark Rolston]], [[Jo Anderson]], [[Rosemary Forsyth]], [[Luoyong Wang]], [[Sakina Jaffrey]], [[Nestor Serrano]], [[Isis Mussenden]], [[Rob Cohen]]
¦ ''[[Rhinestone (film)¦Rhinestone]]'' ¦¦ [[20th Century Fox]] ¦¦ [[Bob Clark]] (director); [[Phil Alden Robinson]], [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[Dolly Parton]], [[Richard Farnsworth]], [[Ron Leibman]], [[Tim Thomerson]], [[Russell Buchanan]], [[Ritch Brinkley]], [[Jesse Welles (actress)¦Jesse Welles]], [[Speck Rhodes]], Steve Peck, Penny Santon, Jerry Potter, Phil Rubenstein, Tony Munafo, Don Hanmer, Guy Fitch, Cindy Perlman
¦ ''[[Staying Alive (1983 film)¦Staying Alive]]'' ¦¦ [[Paramount Pictures]] / [[RSO Records]] ¦¦ [[Sylvester Stallone]] (director/screenplay); [[Norman Wexler]] (screenplay); [[John Travolta]], [[Cynthia Rhodes]], [[Finola Hughes]], [[Steve Inwood]], [[Julie Bovasso]], Charles Ward, [[Norma Donaldson]], Jesse Doran, [[Joyce Hyser]], [[Frank Stallone]], [[Kurtwood Smith]], [[Sylvester Stallone]]
¦ ''[[Nighthawks (1981 film)¦Nighthawks]]'' ¦¦ [[Universal Pictures]] / Martin Poll Productions / The Production Company ¦¦ [[Bruce Malmuth]] (director); [[David Shaber]] (screenplay); [[Sylvester Stallone]], [[Billy Dee Williams]], [[Lindsay Wagner]], [[Persis Khambatta]], [[Nigel Davenport]], [[Rutger Hauer]], [[Hilary Thompson]], [[Joe Spinell]], [[Walter Mathews (actor)¦Walter Mathews]]
¦ ''[[Escape to Victory]]'' ¦¦ [[Paramount Pictures]] / [[Lorimar Television#Theatrical films¦Lorimar]] ¦¦ [[John Huston]] (director); [[Evan Jones (writer)¦Evan Jones]], Yabo Yablonsky, Djordje Milićević, [[Jeff Maguire]] (screenplay); [[Sylvester Stallone]], [[Michael Caine]], [[Max von Sydow]], [[Carole Laure]], [[Benoît Ferreux]], [[Clive Merrison]], [[Maurice Roëves]], [[Michael Cochrane]], [[Zoltán Gera (actor)¦Zoltán Gera]], [[Tim Pigott-Smith]], [[Daniel Massey (actor)¦Daniel Massey]], [[Jean-François Stévenin]], [[Julian Curry]], [[Pelé]], [[Bobby Moore]], [[John Wark]], [[Osvaldo Ardiles]], [[Kazimierz Deyna]], [[Søren Lindsted]], [[Paul Van Himst]], [[Mike Summerbee]], [[Hallvar Thoresen]], [[Russell Osman]], [[Kevin O'Callaghan]], [[Co Prins]], [[Amidou]], [[George Mikell]], [[Gary Waldhorn]], [[Werner Roth (soccer, born 1948)¦Werner Roth]], [[Laurie Sivell]], [[Arthur Brauss]], Michael Wolf, David Shawyer
¦ ''[[Judge Dredd (film)¦Judge Dredd]]'' ¦¦ [[Hollywood Pictures]] / [[Cinergi Pictures]] ¦¦ [[Danny Cannon]] (director); [[William Wisher Jr.]], [[Steven E. de Souza]] (screenplay); [[Sylvester Stallone]], [[Armand Assante]], [[Diane Lane]], [[Rob Schneider]], [[Joan Chen]], [[Jürgen Prochnow]], [[Max von Sydow]], [[James Earl Jones]], [[Adrienne Barbeau]], [[James Remar]], [[Joanna Miles]], [[Balthazar Getty]], [[Maurice Roeves]], [[Ian Dury]], [[Mitchell Ryan]], [[Bradley Lavelle]], Lex Daniel, [[Scott Wilson (actor)¦Scott Wilson]], [[Christopher Adamson (actor)¦Christopher Adamson]], [[Ewen Bremner]], [[Phil Smeeton]], [[Peter Marinker]], [[Angus MacInnes]], [[Mark Moraghan]], [[Al Sapienza]]
¦ ''[[Assassins (1995 film)¦Assassins]]'' ¦¦ [[Warner Bros. Pictures]] / &lt;small&gt;[[Silver Pictures]]&lt;/small&gt; ¦¦ [[Richard Donner]] (director); [[Brian Helgeland]], [[The Wachowskis¦Lilly Wachowski, Lana Wachowski]] (screenplay); [[Sylvester Stallone]], [[Antonio Banderas]], [[Julianne Moore]], Anatoly Davydov, [[Muse Watson]], [[Steve Kahan]], [[Kai Wulff]], Mark Coates, [[Kelly Rowan]], [[Reed Diamond]]
¦ ''[[Rocky]]'' ¦¦ [[United Artists]] / Chartoff-Winkler Productions ¦¦ [[John G. Avildsen]] (director); [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], [[Burgess Meredith]], [[Thayer David]], [[Joe Spinell]], [[Al Silvani]], [[George Memmoli]], [[George O'Hanlon]], [[Stan Shaw]], [[Billy Sands]], [[Pedro Lovell]], [[DeForest Covan]], [[Tony Burton]], [[Shirley O'Hara]], [[Frank Stallone]], [[Lloyd Kaufman]], [[Frank Stallone Jr.]], [[Joe Frazier]], [[Michael Dorn]], [[Arnold Johnson (actor)¦Arnold Johnson]], [[Stu Nahan]], [[Frank Pesce]], [[Chino 'Fats' Williams]]
¦ ''[[F.I.S.T. (film)¦F.I.S.T.]]'' ¦¦ [[United Artists]] / Chateau Productions / Huron Productions Inc. ¦¦ [[Norman Jewison]] (director); [[Joe Eszterhas]], [[Sylvester Stallone]] (screenplay); [[Sylvester Stallone]], [[Rod Steiger]], [[Peter Boyle]], [[Melinda Dillon]], [[David Huffman]], [[Kevin Conway (actor)¦Kevin Conway]], [[Tony Lo Bianco]], [[Cassie Yates]], [[Peter Donat]], [[Frank McRae]], [[Henry Wilcoxon]], [[Richard Herd]], [[Ken Kercheval]], [[James Karen]], [[Stuart Gillard]], [[Brian Dennehy]], [[Nada Rowand]], [[Sam Chew Jr.]], [[John Bleifer]], [[Anthony Kiedis¦Cole Dammett]], [[Bruce McGill]], Tony Mendia, Rozsika Halmos, Elena Karam, John Lehne, Deanne Fator, Jack Slate, [[James Jeter]], Ron Delagardelle, Hugo Bolba, M. Patrick Hughes, Reid Cruickshanks, Chuck Gradi, Earl Montgomery, Vincent Williams
¦ ''[[Paradise Alley]]'' ¦¦ [[Universal Pictures]] / Force Ten Productions Inc. ¦¦ [[Sylvester Stallone]] (director/screenplay); [[Sylvester Stallone]], [[Kevin Conway (actor)¦Kevin Conway]], [[Anne Archer]], [[Joe Spinell]], [[Armand Assante]], Lee Canalito, [[Terry Funk]], [[Frank McRae]], [[Joyce Ingalls]], [[Tom Waits]], Aimee Eccles, [[John Cherry Monks Jr.]], [[Frank Stallone]], [[Ted DiBiase]]
    <redirect title="Sylvester Stallone" />
```



```
1674      <comment>Fixing double redirect to [[Sylvester Stallone]]</comment>
1675      <text bytes="32" sha1="p7gz58bc0f4qqxa7url52a8fje2gfcq" xml:space="preserve">#REDIRECT [[Sylvester Stallone]]</text>
1676 *Wulfgar, the villain in the 1981 Sylvester Stallone film ''[[Nighthawks (1981 film)|Nighthawks]]''
1677 | Co-production with [[Balboa Productions]] and Rocket Science&lt;ref&gt;{{Cite web |last=Kit |first=Borys |date=2023-05-01 |title=Sylvester Stallone Sets 'Cliffhanger' Reboot With 'Angel Has Fallen' Director |url=https://www.hollywoodreporter.com/movies/movie-news/cliffhanger-reboot-set-sylvester-stallone-1235476408/ |access-date=2023-05-02 |website=The Hollywood Reporter |language=en-US}}&lt;/ref&gt; || In development/Pre-production
1678 At age 15, Mendenhall portrayed [[Sylvester Stallone]]'s son in the 1987 sport drama ''[[Over the Top (1987 film)|Over the Top]]'', about a long haul truck driver who tries to win back his alienated son while becoming a champion arm wrestler. That same year, he appeared in the comedy ''[[Going Bananas (film)|Going Bananas]]'' with [[Dom DeLuise]] and [[Jimmie Walker]]. In 1989, Mendenhall worked on ''The Secret of the Ice Cave'' with [[Sally Kellerman]], which was shot in [[Chile]]. All three movies were produced by [[The Cannon Group, Inc.|Cannon Films]]. In 1989, Mendenhall starred opposite [[Christina Applegate]] in the drama ''[[Streets (film)|Streets]]'', once again working with Roger Corman, who produced. On television, Mendenhall had a recurring role for two seasons on the NBC drama ''[[Our House (1986 TV series)|Our House]]'', starring [[Wilford Brimley]].
1679 * 2012 Performed  &quot;[[Eye of the Tiger]]&quot; for [[Sylvester Stallone]] at [[Carmens Banquet Center]] in [[Hamilton, Ontario]].
1680 ''Blood Dragon'' is a stand-alone &quot;expansion pack&quot; based on the world of ''Far Cry 3''. Although ''Blood Dragon'' does not continue the story of ''Far Cry 3'', it shares the same game engine and gameplay mechanics. News of its development was leaked by the Brazilian ratings board, which awarded an 18+ certification based on the title's references to violence, sex, and drugs. The game's retrospective story, characters and visual style are inspired by 1980s action movies, especially those starring [[Arnold Schwarzenegger]], such as ''[[Commando (1985 film)|Commando]]'', ''[[The Terminator]]'', and ''[[Predator (film)|Predator]]'', and [[Sylvester Stallone]], like ''[[Cobra (1986 film)|Cobra]]'', ''[[First Blood]]'', and ''[[Rocky IV]]''. Other film/television references include ''[[RoboCop]]'', ''[[Escape from New York]]'', ''[[Scarface (1983 film)|Scarface]]'' and ''[[Teenage Mutant Ninja Turtles]]'', among others. Pre-orders of the PC version received a digital copy of the soundtrack, produced by Melbourne, Australia-based Power Glove. Due to the success of the title, Ubisoft CEO Yves Guillemot has stated that the game may also get a retail release.
1681 * ''[[The Party at Kitty and Stud's]]'', the debut of [[Sylvester Stallone]]
1682 The 1978 movie ''[[F.I.S.T. (film)|F.I.S.T]]'', tells the same story of Hoffa's beginnings as an organizer and of his rise to power, albeit with more liberties taken. [[Sylvester Stallone]] plays Hoffa as a man with good intentions, dogged on both sides, by both sides of the law.
1683 Pardo will also frequently slip into one of his many characters, including Cajun Jimmy, Plantation Jimmy, Dirty [[Johnny Carson|Carson]], Actor Auditioning for a Blind Role, [[Sylvester Stallone|Stallone]] in a Bottle, [[Chris Russo|Mad Dog Russo]], [[Andrew Dice Clay|Dice]]'s Lady, Larry and Old Man Lizard.
1684 The 1989 film ''[[Lock Up (1989 film)|Lock Up]]'' starring [[Sylvester Stallone]] was shot at the City Work House.&lt;ref&gt;{{cite book | url=https://books.google.com/books?id=kgnS65gAbJoC&amp;pg=PA41 | title=Cincinnati Landmarks | publisher=Arcadia Publishing | date=Oct 29, 2012 | author=Rolfes, Steven | pages=41| isbn=9780738593951 }}&lt;/ref&gt;
1685 Thompson's Fox News prank caught the attention of ''[[The Howard Stern Show]]'', and the impression-dubbed &quot;Fake Arnold&quot; by the show's producers-soon became a mainstay on the program.&lt;ref name=&quot;Dean's Planet Interview&quot;/&gt; At around the same time, Thompson also made a number of appearances on the ''[[The Tim Conway Jr. Show|Conway and Steckler]]'' and ''[[The Tim Conway Jr. Show#Conway and Whitman|Conway and Whitman]]'' radio shows in Los Angeles ([[KNX-FM#KLSX 97.1 Classic Rock|KLSX]]), and [[96fm (Perth radio station)|96fm]]'s ''Breakfast'' show in [[Perth]], Western Australia, where he voiced a number of characters, including Fake Arnold, Robert De Niro, Sylvester Stallone, [[Michael Jackson]], and [[Bill Cosby]].&lt;ref name=&quot;Dean's Planet Interview&quot;/&gt;
1686 |'''''[[Death Race 2000]]'''''||[[Paul Bartel]]||[[David Carradine]], [[Simone Griffeth]], [[Sylvester Stallone]]||United States||[[List of science fiction films of the 1970s|Science fiction adventure]]&lt;ref&gt;{{cite web|url=http://www.fan-de-cinema.com/films/la-course-a-la-mort-de-l-an-2000.html|title=La course à la mort de l'an 2000|publisher=Fan-de-cinema.com|accessdate=9 December 2017}} (in French)&lt;/ref&gt;
1687 |'''''[[Rambo: First Blood Part II]]'''''||[[George P. Cosmatos]], [[Peter MacDonald (director)|Peter MacDonald]]||[[Sylvester Stallone]], [[Richard Crenna]], [[Charles Napier (actor)|Charles Napier]]||United States||War adventure&lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/rambo-first-blood-part-ii-v40204|work=[[AllMovie]]|access-date=August 25, 2018|last=Beday |first=Jeremy|title=Rambo: First Blood Part II}}&lt;/ref&gt;
1688 |'''''[[Rambo III]]'''''||[[Peter MacDonald (director)|Peter MacDonald]]||[[Sylvester Stallone]], [[Richard Crenna]]||United States||War adventure&lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/rambo-iii-v40200|work=[[AllMovie]]|access-date=25 August 2018|last=Williams |first=Karl|title=Rambo III}}&lt;/ref&gt;
1689 |'''''[[Cliffhanger (film)|Cliffhanger]]'''''||[[Renny Harlin]] ||[[Sylvester Stallone]], [[John Lithgow]], [[Janine Turner]]||United States||Mountaineering adventure&lt;ref&gt;{{cite web|url=http://catalog.afi.com/Catalog/moviedetails/59493|work=American Film Institute Catalog|title=Cliffhanger|access-date=8 February 2018|archive-date=9 February 2018|archive-url=https://web.archive.org/web/20180209002739/http://catalog.afi.com/Catalog/moviedetails/59493|url-status=live}}&lt;/ref&gt;
1690 |'''''[[Rambo (2008 film)|Rambo]]'''''||[[Sylvester Stallone]]||[[Sylvester Stallone]], [[Julie Benz]], [[Paul Schulze]]||United States||War adventure&lt;ref&gt;{{cite web|url=http://www.allmovie.com/work/rambo-389422|work=[[AllMovie]]|title=Overview: Rambo|author=Buchanan, Jason|access-date=February 5, 2010}}&lt;/ref&gt;
1691 This is a filmography template for a single actor, [[Sylvester Stallone]]. It is currently transcluded only in the article on the actor (&lt;span class=&quot;plainlinks&quot;&gt;[http://en.wikipedia.org/wiki/Special:Whatlinkshere/Template:Stallone_movies whatlinkshere]&lt;/span&gt;). Single-use templates are unneeded and should exist as text in the main article. '''Subst''' and '''delete''' (and, if possible, get rid of the &quot;v • d • e&quot; links at the top left). - '''[[User:Black Falcon|Black Falcon]]''' &lt;sup&gt;''([[User talk:Black Falcon|Talk]])''&lt;/sup&gt; 17:27, 5 August 2007 (UTC)
1692 In 1972, author [[David Morrell]] published ''[[First Blood (novel)|First Blood]]'', a novel that features a former member of the Army Special Forces named [[John Rambo]]. The novel focuses on the struggle Rambo faces when he attempts to return to civilian life following the end of his tour of duty in Vietnam, and he eventually turns to violence. In 1982, a [[First Blood|film adaptation]] of the novel was released, starring [[Sylvester Stallone]]. The film altered many aspects of the novel, including excising the self-loathing characteristics the protagonist possesses in the novel.&lt;ref&gt;[http://www.davidmorrell.net/faq.cfm] {{webarchive|url=https://web.archive.org/web/20080509133752/http://www.davidmorrell.net/faq.cfm|date=May 9, 2008}}&lt;/ref&gt; The film was a box office success,&lt;ref&gt;{{cite web|url=https://boxofficemojo.com/movies/?id=firstblood.htm |title=First Blood (1982) |publisher=Boxofficemojo.com |access-date=2015-06-03}}&lt;/ref&gt; and spawned a media franchise.
1693 * In the film ''[[First Blood]]'' (1982), [[Sylvester Stallone]] portrays [[John Rambo]], a former Green Beret searching for his old friend from his time in the Special Forces. The film is based on the 1972 novel ''[[First Blood (novel)|First Blood]]'' by [[David Morrell]]. The film altered many aspects of the novel, including excising the self-loathing characteristics the protagonist possesses in the novel.&lt;ref&gt;[http://www.davidmorrell.net/faq.cfm] {{webarchive|url=https://web.archive.org/web/20080509133752/http://www.davidmorrell.net/faq.cfm|date=May 9, 2008}}&lt;/ref&gt; The film was a box office success,&lt;ref&gt;{{cite web|url=https://boxofficemojo.com/movies/?id=firstblood.htm |title=First Blood (1982) |publisher=Boxofficemojo.com |access-date=2015-06-03}}&lt;/ref&gt; and spawned a [[Rambo (franchise)|media franchise]].
1694 &quot;Crystal Blue Persuasion&quot; has appeared in the films ''[[A Walk on the Moon]]'' (1999), ''The Secret Life of Girls'' (1999), ''[[Zodiac (film)|Zodiac]]'' (2007), ''[[The Nanny Diaries]]'' (2007), and the TV show ''[[How to Make it in America]]'' (2010). The song is also featured in the movie ''[[The Expendables 2]]'' (2012), while Barney ([[Sylvester Stallone]]) is flying his plane. It was played during the pool scene in the TV movie ''[[Growing Up Brady (film)|Growing Up Brady]],'' and was in the pilot episode of ''[[The Wonder Years]]''.&lt;ref&gt;{{Cite web|url=https://www.tunefind.com/show/the-wonder-years/season-1/48102|title=Music from The Wonder Years S1E01|website=Tunefind|language=en-us|access-date=2019-10-17}}&lt;/ref&gt;
1695 |''[[Oscar (1991 film)|Oscar]]''||{{sortname|John|Landis}}||[[Sylvester Stallone]], [[Ornella Muti]], [[Peter Riegert]]||United States||&lt;ref&gt;{{cite web|url=http://www.allmovie.com/movie/oscar-v36689|publisher=AllMovie|title=Oscar|last=Deming|first=Mark|access-date=June 29, 2015}}&lt;/ref&gt;
1696 |''[[Stop! Or My Mom Will Shoot!]]''||{{sortname|Roger|Spottiswoode}}||[[Sylvester Stallone]], [[Estelle Getty]], [[JoBeth Williams]]||United States||&lt;ref&gt;{{cite web|url=http://www.allmovie.com/movie/stop!-or-my-mom-will-shoot-v46981|publisher=AllMovie|title=Stop! Or My Mom Will Shoot!|last=Deming|first=Mark|access-date=July 2, 2015}}&lt;/ref&gt;
1697 |''[[Demolition Man (film)|Demolition Man]]''||[[Marco Brambilla]]||[[Sylvester Stallone]], [[Wesley Snipes]], [[Sandra Bullock]]||United States||Science fiction action comedy&lt;ref&gt;{{Cite web | url=https://www.allmovie.com/movie/demolition-man-v119000 |title = Demolition Man (1993) - Marco Brambilla {{pipe}} Synopsis, Characteristics, Moods, Themes and Related}}&lt;/ref&gt;
1698 Created especially for the film ''[[Midnight Express (film)|Midnight Express]]'', [[Alan Parker]], the director of the film, explicitly asked Moroder for a song in the style of &quot;[[I Feel Love]]&quot;, which Moroder composed for [[Donna Summer]]. It was Moroder's second time composing a movie soundtrack after his work on 1972's German softcore sex film &quot;Sex Life in a Convent&quot;.&lt;ref&gt;{{Cite web|title=Giorgio Moroder, Filmkomponist by Thomas Staedeli|url=https://www.cyranos.ch/skmoro-d.htm|access-date=2021-08-30|website=www.cyranos.ch}}&lt;/ref&gt;&lt;ref&gt;{{Citation|title=Sex Life in a Convent (1972) - IMDb|url=https://www.imdb.com/title/tt0068377/|language=en-US|access-date=2021-08-30}}&lt;/ref&gt;&lt;ref&gt;[https://www.theguardian.com/music/2013/oct/31/giorgio-moroder-sylvester-stallone-bob-dylan-donna-summer &quot;Giorgio Moroder: 'Sylvester Stallone wanted Bob Dylan to sing on a Rambo movie'&quot;], The Guardian, 31 October 2013.&lt;/ref&gt; The song's main melody was played on a [[Roland Corporation|Roland]] [[Roland SH-1000|SH-2000]] synthesizer, while the [[bass line]]s were played on a [[Minimoog]] [[synthesizer]]. The track also has a [[flanging]] effect produced by the MXR Flanger, while other instruments used include an [[ARP String Ensemble|ARP/Solina String Ensemble]], [[Fender Rhodes]], [[Hohner Clavinet]], and [[piano]].&lt;ref&gt;[http://www.dolphinmusic.co.uk/article/5916-giorgio-moroder-gear-guide-inc-new-novation-morodernova-.html Giorgio Moroder Gear Guide], [[Dolphin Music]]&lt;/ref&gt;
1699 ''Mr. Insincere'', (written / performed by Howson)&lt;ref name=&quot;APRA&quot; /&gt; appears in the Disney movie ''[[Burn Hollywood Burn]]'' (1998), starring [[Eric Idle]] (as [[Alan Smithee]]) and [[Ryan O'Neal]]; with [[Whoopi Goldberg]], [[Jackie Chan]], [[Sylvester Stallone]], and [[Robert Evans (producer)|Robert Evans]].&lt;ref&gt;{{cite web | title=IMDb entry on ''An Alan Smithee Film: Burn Hollywood Burn'' (1998)|website= [[Internet Movie Database|IMDb]] |url= https://www.imdb.com/title/tt0118577/fullcredits#cast |accessdate= 21 September 2007 }}&lt;/ref&gt; Howson was asked by [[Steven Berkoff]] to direct the TV version of Berkoff's international hit theatre show ''[[Shakespeare's Villains]]'' (2002).&lt;ref name=&quot;IMDb&quot; /&gt;
1700 In January, Reading could not have got a better start. On New Year's Day, Reading put six past West Ham United. On 6 January, Reading were due to play Burnley in the FA Cup but it was postponed due to a waterlogged pitch, but Reading beat Burnley on the Tuesday night, a 3-2 win. Reading went to Everton unbeaten in the new year, and the game ended 1-1 in a match attended by [[Sylvester Stallone]] as an Everton supporter and guest of the chairman. Reading played Sheffield United at 2-0 Reading came a push-up in the dug out. Warnock and Downs sent off. Reading won 3-1. Reading Played Birmingham in the FA Cup (again). Birmingham had just enjoyed a 5-1 win at Newcastle United but they lost 3-2 to the Royals. Reading beat Wigan by the same score.
1701 | [[Sylvester Stallone]] || ''[[Creed (film)|Creed]]'' || [[Rocky Balboa]]
1702 [[Category:Images of Sylvester Stallone]]</text>
1703 In 2012, Adele and then-partner Konecki purchased a $3.4&amp;nbsp;million [[Art Deco#Architecture|Art Deco]] villa in [[Portslade]], on the outskirts of [[Brighton and Hove]], which she sold for $3.7&amp;nbsp;million in 2016. That same year, she bought two houses built side-by-side in [[Kensington]] for $7.7&amp;nbsp;million and $7.3&amp;nbsp;million, respectively, with the intention of
```



```
combining them. Adele also bought a home for her mother in [[West End of London|West London]] for around $817,000.&lt;ref name=&quot;Joyce2022&quot;&gt;{{Cite magazine |last=Chen |first=Joyce |date=11 March 2022 |title=Adele's Houses: Inside Her Impressive Real Estate Portfolio |url=https://www.architecturaldigest.com/story/adeles-houses-inside-her-impressive-real-estate-portfolio |url-status=dead |magazine=[[Architectural Digest]] |issn=0003-8520 |archive-url=https://web.archive.org/web/20220424163751/https://www.architecturaldigest.com/story/adeles-houses-inside-her-impressive-real-estate-portfolio |archive-date=24 April 2022 |access-date=9 July 2022}}&lt;/ref&gt; In 2013, she temporarily rented [[Paul McCartney]]'s {{convert|12,000|sqfoot|m2|adj=on}} former mansion for an undisclosed price. In 2015, Adele purchased a $5.2&amp;nbsp;million [[Mediterranean Revival architecture|Mediterranean-style]] vacation mansion in [[Malibu, California]], and sold it in 2017 for less than its original purchase price, $4.8&amp;nbsp;million.&lt;ref name=&quot;Joyce2022&quot; /&gt; She envisioned buying several properties on the same street in a Beverly Hills [[gated community]], beginning with the first home purchased for $9.5&amp;nbsp;million from [[Don Mischer]] in 2016.&lt;ref&gt;{{Cite news |last=Casey |first=Brendan |date=24 November 2021 |title=Inside Adele's $41.5 million Beverly Hills real estate portfolio made up of adjoining houses |publisher=[[REA Group]] |url=https://www.realestate.com.au/news/inside-adeles-415-million-beverly-hills-real-estate-portfolio-made-up-of-adjoining-houses/ |url-status=dead |access-date=9 July 2022 |archive-url=https://web.archive.org/web/20211220014219/https://www.realestate.com.au/news/inside-adeles-415-million-beverly-hills-real-estate-portfolio-made-up-of-adjoining-houses/ |archive-date=20 December 2021}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Kiefer |first=Peter |date=16 September 2016 |title=Meet the Former MTV VJ Who Now Sells Multimillion-Dollar Homes to Adele and Emilia Clarke |work=[[The Hollywood Reporter]] |url=https://www.hollywoodreporter.com/lifestyle/lifestyle-news/meet-mtv-vj-who-sells-928227/ |url-status=dead |access-date=10 July 2022 |archive-url=https://web.archive.org/web/20220710144239/https://www.hollywoodreporter.com/lifestyle/lifestyle-news/meet-mtv-vj-who-sells-928227/ |archive-date=10 July 2022}}&lt;/ref&gt; During her Madison Square Garden tour in September 2016, Adele rented NBA player [[Deron Williams]]' [[Tribeca]] apartment for three weeks at a monthly rent of $60,000.&lt;ref&gt;{{Cite news |last=Lucking |first=Liz |date=20 December 2017 |title=Former NBA Star Deron Williams Relists Manhattan Penthouse for $26 Million |work=Mansion Global |url=https://www.mansionglobal.com/articles/former-nba-star-deron-williams-relists-manhattan-penthouse-for-26-million-83918 |url-status=dead |access-date=9 July 2022 |archive-url=https://web.archive.org/web/20211208214004/https://www.mansionglobal.com/articles/former-nba-star-deron-williams-relists-manhattan-penthouse-for-26-million-83918 |archive-date=8 December 2021}}&lt;/ref&gt;&lt;ref name=&quot;Joyce2022&quot; /&gt; In 2017, she and Konecki purchased a $5.3&amp;nbsp;million [[Tudor Revival architecture|Tudor]] mansion called Ridge Hill Manor, located in the English countryside on the outskirts of [[East Grinstead]].&lt;ref name=&quot;Joyce2022&quot; /&gt; In 2019 and 2021, Adele bought two more Beverly Hills mansions for $10.65&amp;nbsp;million and $10&amp;nbsp;million, respectively; the latter was purchased from [[Nicole Richie]] and her husband, [[Joel Madden]].&lt;ref&gt;{{Cite magazine |last=Chen |first=Joyce |date=17 May 2019 |title=Adele Buys a $10.65 Million Beverly Hills Mansion |url=https://www.architecturaldigest.com/story/adele-buys-a-new-beverly-hills-mansion |url-status=dead |magazine=Architectural Digest |issn=0003-8520 |archive-url=https://web.archive.org/web/20220709175446/https://www.architecturaldigest.com/story/adele-buys-a-new-beverly-hills-mansion |archive-date=9 July 2022 |access-date=9 July 2022}}&lt;/ref&gt;&lt;ref&gt;{{Cite magazine |last=Dweck |first=Sophie |date=17 June 2021 |title=Adele Buys Next-Door Neighbor Nicole Richie's Beverly Hills House for $10 Million |url=https://www.architecturaldigest.com/story/adele-buys-next-door-neighbor-nicole-richies-beverly-hills-house-for-dollar10-million |url-status=dead |magazine=Architectural Digest |issn=0003-8520 |archive-url=https://web.archive.org/web/20220710144240/https://www.architecturaldigest.com/story/adele-buys-next-door-neighbor-nicole-richies-beverly-hills-house-for-dollar10-million |archive-date=10 July 2022 |access-date=9 July 2022}}&lt;/ref&gt; In February 2022, it was reported she bought [[Sylvester Stallone]]'s {{convert|21,000|sqfoot|m2|adj=on}} former mansion in [[Beverly Park, Los Angeles]], for $58&amp;nbsp;million.&lt;ref&gt;{{Cite news |last=Flemming |first=Jack |date=6 February 2021 |title=Hot Property: Sylvester Stallone lists Beverly Park trophy estate for $110 million |work=[[Los Angeles Times]] |url=https://www.latimes.com/business/real-estate/story/2021-02-06/sylvester-stallone-beverly-park-matthew-perry-malibu-kelly-clarkson-encino |url-status=dead |access-date=10 July 2022 |archive-url=https://web.archive.org/web/20220619160759/https://www.latimes.com/business/real-estate/story/2021-02-06/sylvester-stallone-beverly-park-matthew-perry-malibu-kelly-clarkson-encino |archive-date=19 June 2022}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Flemming |first=Jack |date=25 February 2022 |title=Adele buys Sylvester Stallone's Beverly Park mansion for $58 million - a deep discount |work=Los Angeles Times |url=https://www.latimes.com/business/real-estate/story/2022-02-25/sylvester-stallone-sells-beverly-park-mansion-for-58-million-a-deep-discount |url-status=dead |access-date=10 July 2022 |archive-url=https://web.archive.org/web/20220714171911/https://www.latimes.com/business/real-estate/story/2022-02-25/sylvester-stallone-sells-beverly-park-mansion-for-58-million-a-deep-discount |archive-date=14 July 2022}}&lt;/ref&gt;
1704 Morrell's novel was popular in its time and was the inspiration for the hugely successful 1982 film ''[[First Blood]]'' starring [[Sylvester Stallone]], released to international acclaim.
1705 |RTitle=[[Sylvester Stallone]]
1706 |'''''[[Nighthawks (1981 film)|Nighthawks]]'''''||[[Bruce Malmuth]]||[[Sylvester Stallone]], [[Rutger Hauer]], [[Billy Dee Williams]]||United States||Crime thriller, action thriller&lt;ref&gt;{{cite web|url=http://allmovie.com/work/nighthawks-35341|title=Overview:Nighthawks|author=Rasmussen, Linda|accessdate=July 20, 2010|work=Allmovie}}&lt;/ref&gt;
1707 |'''''[[First Blood]]'''''||[[Ted Kotcheff]]||[[Sylvester Stallone]], [[Richard Crenna]], [[Brian Dennehy]]||United States||Action thriller&lt;ref&gt;{{cite web|url=http://allmovie.com/work/first-blood-17472|title=Overview:First Blood|author=Erickson, Hal|accessdate=July 21, 2010|work=Allmovie}}&lt;/ref&gt;
1708 |'''''[[Rambo: First Blood Part II]]'''''||[[George Pan Cosmatos]]||[[Sylvester Stallone]], [[Richard Crenna]], [[Charles Napier (actor)|Charles Napier]] ||United States||Action thriller&lt;ref&gt;{{cite web|url=http://allmovie.com/work/rambo-first-blood-part-ii-40204|title=Overview:Rambo: First Blood Part II|author=Beday, Jeremy|accessdate=July 22, 2010|work=Allmovie}}&lt;/ref&gt;
1709 |'''''[[Cobra (1986 film)|Cobra]]'''''||[[George Pan Cosmatos]]||[[Sylvester Stallone]], [[Brigitte Nielsen]], [[Reni Santoni]] ||United States||Action thriller&lt;ref&gt;{{cite web|url=http://allmovie.com/work/cobra-10123|title=Overview:Cobra|author=Brenner, Paul|accessdate=July 23, 2010|work=Allmovie}}&lt;/ref&gt;
1710 |'''''[[Rambo III]]'''''||[[Peter MacDonald (film director)|Peter MacDonald]]||[[Sylvester Stallone]], [[Richard Crenna]], [[Kurtwood Smith]] ||United States||Action thriller&lt;ref&gt;{{cite web|url=http://allmovie.com/work/rambo-iii-40200|title=Overview:Rambo III|author=Williams, Karl|accessdate=July 25, 2010|work=Allmovie}}&lt;/ref&gt;
1711 |'''''[[Cliffhanger (film)|Cliffhanger]]'''''||[[Renny Harlin]]||[[Sylvester Stallone]], [[John Lithgow]], [[Michael Rooker]], [[Janine Turner]] ||United States||Action thriller&lt;ref&gt;{{cite web|url=http://allmovie.com/work/cliffhanger-9983|title=Cliffhanger: Overview|author=Brenner, Paul|accessdate=July 6, 2010}}&lt;/ref&gt;
1712 |'''''[[The Specialist]]'''''||[[Luis Llosa]]||[[Sylvester Stallone]], [[Sharon Stone]], [[James Woods]], [[Rod Steiger]], [[Eric Roberts]] ||United States||Action thriller&lt;ref&gt;{{cite web|url=http://www.allmovie.com/work/the-specialist-133419|work=Allmovie|title=The Specialist: Overview|author=Williams, Karl|accessdate=July 7, 2010}}&lt;/ref&gt;
1713 |'''''[[Assassins (1995 film)|Assassins]]'''''||[[Richard Donner]]||[[Sylvester Stallone]], [[Antonio Banderas]], [[Julianne Moore]]||United States||&lt;ref&gt;{{cite web|url=http://allmovie.com/work/assassins-135824|work=Allmovie|title=Assassins: Overview|accessdate=September 18, 2009|author=Kaye, Don}}&lt;/ref&gt;
1714 |'''''[[Get Carter (2000 film)|Get Carter]]'''''||Stephen Kay||[[Sylvester Stallone]]||United States||&lt;ref&gt;{{cite web|url=http://www.allmovie.com/work/get-carter-214069|work=Allmovie|author=Deming, Mark|accessdate=June 25, 2010|title=Get Carter: Overview}}&lt;/ref&gt;
1715 |'''''[[Rambo (2008 film)|Rambo]]'''''||[[Sylvester Stallone]]||[[Sylvester Stallone]], [[Julie Benz]], [[Matthew Marsden]]||United States||Action thriller&lt;ref&gt;{{cite web|url=http://allmovie.com/work/rambo-389422|work=Allmovie|author=Buchanan, Jason|accessdate=June 29, 2010|title=Rambo: Overview}}&lt;/ref&gt;
1716 &lt;ref&gt;[http://www.festivalvault.ca/pdf/psychobeachparty-mk.pdf Psycho Beach Party (pdf file)]{{Dead link|date=December 2019 |bot=InternetArchiveBot |fix-attempted=yes }}&lt;/ref&gt;&lt;ref&gt;[http://www.jeremywalker.com/pages/films/imagesnote/wonderland_notes.doc Wonderland (doc file)]&lt;/ref&gt; His motion picture credits include [[Sylvester Stallone]]'s ''Rocky Balboa'' and ''John Rambo'', [[William Friedkin]]'s ''Bug'', [[Eli Roth]]'s ''Hostel'' and ''Cabin Fever'', [[David Jacobson (director)|David Jacobson]]'s ''Down in the Valley'' and [[James Cox (director)|James Cox]]'s ''Wonderland''.
1717 | ''[[Rambo (2008 film)|Rambo]]'' || [[Sylvester Stallone]] ||
1718 | [[2006 in film|2006]] || ''[[Rocky Balboa (film)|Rocky Balboa]]'' || [[Sylvester Stallone]] ||
1719 [[Category:Images of Sylvester Stallone]]
1720 | style=&quot;text-align:center;&quot; | 2015 || '''[[Sylvester Stallone]]''' || ''[[Creed (film)|Creed]]'' || [[Rocky Balboa]]
1721 '''Richard Lee Zumwalt Jr.''' (September 24, 1951 - March 19, 2003) was an American professional [[arm-wrestler]] and actor. He is known for playing the character of Bob &quot;Bull&quot; Hurley in the 1987 [[Sylvester Stallone]] movie ''[[Over the Top (1987 film)|Over the Top]]''.
1722 | Robert Redford || [[Sylvester Stallone]] || Burt Reynolds || Burt Reynolds || Burt Reynolds
1723 | Burt Reynolds || Woody Allen || Woody Allen || Sylvester Stallone || John Travolta
1724 | Burt Reynolds || Burt Reynolds || Clint Eastwood || Clint Eastwood || Sylvester Stallone
1725 | [[Dudley Moore]] || Sylvester Stallone || Sylvester Stallone || Harrison Ford || Clint Eastwood
1726 | Sylvester Stallone || Tom Cruise || [[Danny DeVito]] || [[Michael J. Fox]] || [[Kevin Costner]]
1727 | [[Whoopi Goldberg]] || Sylvester Stallone || [[Robin Williams]] || Mel Gibson || Sean Connery
1728 | Mel Gibson || Robin Williams || [[Sylvester Stallone]] || [[Tommy Lee Jones]] || Kevin Costner&lt;br&gt;[[Michael Douglas]] (tie)
1729 | Sylvester Stallone
1730 | Sylvester Stallone
1731 | Sylvester Stallone
1732 | Sylvester Stallone
1733 | Sylvester Stallone
1734 | Sylvester Stallone
1735 | Sylvester Stallone
1736 | Sylvester Stallone
1737 | Sylvester Stallone
1738 | Sylvester Stallone
1739 | Sylvester Stallone
1740 | Sylvester Stallone
1741 | Sylvester Stallone
1742 | Sylvester Stallone
1743 In 1981, [[Michael Caine]] and [[Sylvester Stallone]] starred in the film ''[[Escape to Victory]]'', directed by [[John Huston]], which told the story of a group of [[Allies (World War II)|Allied]] POWs who are challenged to a match against the prison's guards. While the film's POWs are not Ukrainian but rather predominately Westerners, the story parallels are clear: they are threatened with death if they win, the playing ground is surrounded with Nazi guards and attack dogs, the referee ignores vicious and brutal fouls committed by the German team, yet the Allied prisoner team ignore the threat and draw the match, thus risking forfeiting their lives. (Huston's film has a ''[[deus ex machina]]'' ending which conflicts with the original Soviet story when the spectators storm the
```



field at the match's end and the POWs escape in the resultant confusion, but as no event similar to this actually occurred in the West during World War II, it is generally assumed that this film was inspired by the legendary/propaganda version of the Death Match.)

1744 ¦ year= 2006¦ title= British Hit Singles &amp; Albums¦ edition= 19th¦ publisher= Guinness World Records Limited ¦ location= London¦ isbn= 1-904994-10-5¦ page= 20}}&lt;/ref&gt; 1995's &quot;Real&quot;, her last US chart hit, was taken from the [[soundtrack]] to the [[Sylvester Stallone]] film ''[[The Specialist]]''.

1745 ¦''[[Demolition Man (film)¦Demolition Man]]''¦¦[[Marco Brambilla]]¦¦[[Sylvester Stallone]], [[Wesley Snipes]], [[Sandra Bullock]], [[Nigel Hawthorne]]¦¦United States¦¦

1746 ¦''[[Judge Dredd (film)¦Judge Dredd]]''¦¦[[Danny Cannon]]¦¦[[Sylvester Stallone]], [[Armand Assante]], [[Diane Lane]], [[Rob Schneider]]¦¦United States¦¦

1747 Proietti was also a voice dubber of films and television shows into the Italian language. He has dubbed the voices of actors such as [[Robert De Niro]], [[Sylvester Stallone]], [[Richard Burton]], [[Richard Harris]], [[Dustin Hoffman]], [[Paul Newman]], [[Charlton Heston]] and [[Marlon Brando]].&lt;ref name=bioAA /&gt; His credits also include the role of the [[Genie (Disney)¦Genie]] in the Italian version of the ''[[Aladdin (franchise)¦Aladdin]]'' film series and Draco in ''[[Dragonheart]]''.&lt;ref name=bioAA /&gt;&lt;ref&gt;{{cite web¦url=https://cinema.fanpage.it/dopo-la-morte-di-gigi-proietti-il-genio-di-aladdin-e-rimasto-senza-voce/ ¦title= Dopo la morte di Gigi Proietti, il Genio di Aladdin è rimasto senza voce ¦publisher=cinema.fanpage.it¦language=it¦ date= 2 November 2020¦access-date=3 November 2020}}&lt;/ref&gt; He also provided the Italian voice of [[Gandalf]] in ''[[The Hobbit (film series)¦The Hobbit]]'' film series, replacing the late [[Gianni Musy]], who dubbed [[Gandalf]] in ''[[The Lord of the Rings (film series)¦The Lord of the Rings]]'', as well as [[Sylvester the Cat¦Sylvester]] from ''[[Looney Tunes]]'' during the 1960s.

1748 ¦rowspan=&quot;2&quot;¦[[Sylvester Stallone]]

1749 ¦''[[Death Race 2000]]''¦¦[[Paul Bartel]]¦¦[[David Carradine]], [[Sylvester Stallone]], [[Simone Griffeth]]¦¦United States¦¦Action Comedy Sport

1750 Track 10, the rock song &quot;Spectacular&quot;, featured in a [[Pepsi]] commercial.&lt;ref&gt;{{Cite web ¦title=专辑：王菲--《只爱陌生人》 ¦url=https://ent.sina.com.cn/y/d/2005-05-27/1412736042.html ¦access-date=April 20, 2024 ¦website=ent.sina.com.cn ¦language=zh}}&lt;/ref&gt; A VCD showing the filming of the commercial was released with some versions of the album. The title track, &quot;Only Love Strangers&quot;, was featured in the [[Sylvester Stallone]] film ''[[Get Carter (2000 film)¦Get Carter]]''.&lt;ref&gt;[http://ent.163.com/edit/001019/001019_63865.html Faye Wong &quot;Only Love Strangers&quot;, a Stallone film episode], [[163.com]], October 19, 2000 {{in lang¦zh}}&lt;/ref&gt; Track 5 was used by Taiwan's [[STAR Chinese Channel]] as the ending theme song of the Chinese television series ''[[The Taiping Heavenly Kingdom (TV series)¦The Taiping Heavenly Kingdom]]''. The original ending theme song was sung by [[Mao Amin]].

1751 [[Baptista]] and [[Roberto Carlos]], [[Rudy Fernández (basketball)¦Rudy Fernández]], [[Ona Carbonell]], [[Carlos Sainz Sr.¦Carlos Sainz]], [[Carlos Sainz Jr.]], [[Novak Djokovic]], [[Fernando Alonso]], [[Roberto Bautista]] and [[Rafa Nadal]], [[Carlos Alcaraz]] motorcycle [[Toni Elías]], [[Jorge Lorenzo]], [[Joan Mir]], [[Marc Marquez]] and [[Àlex Márquez¦Alex Marquez]], the politicians [[Albert Rivera]], [[Mariano Rajoy]], [[Pablo Iglesias Turrión¦Pablo Iglesias]], [[Pedro Sánchez]] and [[Miguel Angel Revilla]], the models [[Cara Delevingne]], [[Sara Sampaio]], [[Karlie Kloss]], [[Alessandra Ambrosio]], [[Heidi Klum]] and [[Adriana Karembeu]], the astronaut [[Charles Duke]], the writer [[Dan Brown]], the dog trainer [[Cesar Millan]], actors [[Will Smith]], [[Jaden Smith]], [[Jada Pinkett Smith]], [[Jackie Chan]], [[Margot Robbie]], [[Mena Massoud]], [[Naomi Scott]], [[Martin Lawrence]]; ''[[Lost (2004 TV series)¦Lost]]'' stars [[Matthew Fox]] and [[Jorge Garcia]]; ''[[House (TV series)¦House M.D.]]'' stars [[Olivia Wilde]] and [[Lisa Edelstein]]; [[Hugh Jackman]], [[Rob Schneider]], [[Matt Dallas]], [[Sylvester Stallone]], [[Hugh Grant]], [[Michael C. Hall]], [[Jean Reno]], [[Jason Biggs]], [[Seann William Scott]], [[Geraldine Chaplin¦Geraldine]], [[Oona Castilla Chaplin]], [[Johnny Knoxville]], [[Tom Cruise]], [[Annabelle Wallis]], [[Sofia Boutella]], [[Emily Blunt]], [[Ana de Armas]], [[Cameron Diaz]], [[Nikolaj Coster-Waldau]], [[Gary Dourdan]], [[Eva Mendes]], [[Jennifer Connelly]], [[Steve Carell]], [[Kristen Wiig]], [[Tim Burton]], [[Jamie Lee Curtis]] [[Charlize Theron]], [[Kristen Stewart]], [[Robert Pattinson]], [[Taylor Lautner]], [[Zoe Kravitz]], [[Clive Owen]], [[Alec Baldwin]], [[Sam Heughan]], [[Ricardo Darín]], [[Elisabeth Moss]], [[Jamie Bell]], [[Reese Witherspoon]], [[Laura Natalia Esquivel]], [[Robert Englund]], [[Danny DeVito]], [[Rami Malek]], [[Mark Wahlberg]], [[Ben Affleck]], [[Oscar Isaac]], [[Norman Reedus]], [[Jeffrey Dean Morgan]], [[Jennifer Aniston]], [[Diego Luna]], [[Denzel Washington]], [[Daniel Craig]], [[Dolph Lundgren]], [[Jeremy Irons]], [[Freddie Highmore]], [[Ryan Reynolds]], [[Richard Gere]], [[Vincent Cassel]], [[Valentina Zenere]], [[Daniel Radcliffe]], [[Rupert Grint]], the [[James and Oliver Phelps¦Phelps twins]], [[Eddie Redmayne]], [[Liam Neeson]], [[Callum Turner]], [[Russell Crowe]], [[Dwayne Johnson]], [[Kevin Hart (actor)¦Kevin Hart]], [[Esmeralda Pimentel]], [[Oscar Jaenada]], [[Mario Casas]], [[Amaia Salamanca]], [[Blanca Suarez]], [[Jose Coronado]], [[Isaac Hempstead Wright]], [[Liam Cunningham]], [[Liam Hemsworth]], [[Josh Hutcherson]] and [[Jennifer Lawrence]], [[William Levy (actor)¦William Levy]], [[Olivia Munn]], [[Boyd Holbrook]], [[María Eugenia Suárez¦China Suarez]], [[Benicio del Toro]], [[Tim Robbins]], [[Jessica Chastain]], [[Adriana Karembeu]], [[Ed Harris]], [[Jake Gyllenhaal]], [[Yahya Abdul-Mateen II]], [[Kelsey Grammer]], [[Matthew McConaughey]], [[Woody Allen]], [[Ernesto Alterio]], [[Peter Coyote]], [[Nicole Kidman]], [[Elsa Pataky]], ''[[Marvel Cinematic Universe]]'' stars [[Chris Hemsworth]], [[Tom Holland (actor)¦Tom Holland]], [[Benedict Cumberbatch]], [[Chris Pratt]] and [[Bradley Cooper]], [[Gwyneth Paltrow]], [[Debi Mazar]], [[Patrick Stewart]], [[Bryan Cranston]], [[Peter Dinklage]], [[Alex Pettyfer]] and [[Teresa Palmer]], [[Channing Tatum]], [[Eva Longoria]], [[Orlando Bloom]], [[Vin Diesel]], [[Dakota Johnson]], [[Sofía Vergara]], [[Leo Bassi]], [[Kenneth Branagh¦Kennet Branagh]], [[Richard Madden]], [[Willem Dafoe]], [[Sigourney Weaver]], [[Martin Sheen]], [[Emilio Estevez]], [[Ed Skrein]], [[Mel Gibson]], Johhny Depp[[David Hasselhoff]], [[Elijah Wood]], [[Sasha Grey]], [[Tom Hanks]], [[Mariana Treviño]], [[Sharon Stone]], [[Harrison Ford]], [[Drew Barrymore]], [[Timothy Olyphant]], ''[[Stranger Things]]'' stars [[David Harbour]], [[Millie Bobby Brown]] and [[Noah Schnapp]], [[Oscar Martínez (actor)¦Óscar Martínez]], [[Penelope Cruz]], [[Pedro Almodóvar]], [[Antonio Banderas]], [[Salma Hayek]], [[Rowan Atkinson]], [[Amanda Seyfried]], [[Lorenzo Lamas]], [[Emma Stone]], [[Andrew Garfield]], [[Jesse Eisenberg]] and [[Justin Timberlake]], singers [[Jonas Brothers]], [[Enrique Iglesias]], [[Backstreet Boys]], [[McFly]], [[Tokio Hotel]], [[One Direction]], [[Bizarrap]], [[David Bisbal]], [[Álvaro Soler]], [[India Martinez]], [[La Oreja De Van Gogh]], [[Carlos Baute]], [[Kylie Minogue]], [[Alejandro Sanz]], [[Shakira]], [[Alesha Dixon]], [[Ashley Tisdale]], [[Ke$ha]], [[Iggy Pop]], [[Anastacia]], [[Chayanne]], [[Isabel Pantoja]], [[Emilia de Poret]], [[Julieta Venegas]], [[Norah Jones]], [[Pablo Alboran]], [[Camilo (singer)¦Camilo]], [[Ava Max]], [[Eros Ramazzotti]], [[Monica Naranjo]], [[Chris Isaak]], [[Anitta (singer)¦Anitta]], [[Salvador Sobral]], [[Rosalía]], [[Milow]], [[Craig David]], [[Emeli Sandé]], [[Fleur East]], [[Cecilia Bartoli]], [[Michel Teló]], [[Lang Lang]], [[Miguel Bosé]], [[Carlos Vives]], [[Amaia Montero]], [[David Summers Rodríguez¦David Summers]] ([[Hombres G]]), [[Jason Derulo]], [[Kate Ryan]], [[Ricky Martin]], [[Austin Mahone]], [[La Quinta Estacion]], [[Tini (singer)¦Tini]], [[Danna Paola]], [[Alicia Keys]], [[Pitbull (rapper)¦Pitbull]], [[Belinda Peregrín¦Belinda]], [[David Guetta]], [[Raffaella Carrà]], [[Carla Bruni]], [[Bob Sinclar]], [[Ella Baila Sola]], [[James Blunt]], [[Meghan Trainor]], [[Raphael (singer)¦Raphael]], [[Sebastian Yatra]], [[Ed Sheeran]], [[Lali Esposito]], [[Billy Ray Cyrus]], [[Pimpinela]], [[Luis Fonsi]], [[Melendi]], [[Emilia Mernes¦Emilia]], [[María Becerra]], [[Juanes]], [[Manuel Turizo]], [[Jeanette (Spanish singer)¦Jeanette]], [[Nathy Peluso]], [[Sergio Dalma]], [[Gloria Estefan]], [[Maná (band)¦Maná]], [[The Baseballs]], [[Juan Luis Guerra]], [[Nelly Furtado]], [[Lele Pons]], [[Guaynaa]],[[Lenny Kravitz]], [[Duffy (singer)¦Duffy]], [[30 Seconds to Mars]] members [[Jared Leto]]; [[Taylor Swift]], [[Romeo Santos]], [[Andres Calamaro]], [[La Toya Jackson]], [[Carlos Santana]], [[Michael Buble]], [[LMFAO]], [[Laura Pausini]], [[Mika (singer)¦Mika]], [[King Africa]], [[Christopher Cross]], [[Jose Luis Perales]], [[Karol G]], [[Jamie Cullum]], [[Paulina Rubio]], [[Carly Rae Jepsen]], 80s singers [[Rick Astley]], [[Spandau Ballet]], [[Bonnie Tyler]] and [[Ana Torroja]], reggaeton singers [[Daddy Yankee]], [[Maluma]], [[Ozuna (singer)¦Ozuna]], [[Rauw Alejandro]], [[Duki (rapper)¦Duki]], [[Nicky Jam]] and [[J Balvin]] productor music [[Manuel Alejandro]] young actors [[Dafne Keen]], [[Julio Peña Fernández¦Julio Peña]] and [[Clara Galle]] singers [[Vanessa Hudgens]], [[Selena Gomez]], [[Alfred Garcia]], [[Lola Indigo]], [[Nicki Nicole]], [[Ana Mena]], [[Becky G]], [[Ana Guerra]], [[Aitana (singer)¦Aitana]], [[Justin Bieber]] and [[Miley Cyrus]] and comedias [[Adam Sandler]], [[Frat Pack]] Members [[Ben Stiller]], [[Owen Wilson]] and [[Will Ferrell]], [[Santiago Segura]], [[Florentino Fernández (actor)¦Florentino Fernández]], [[José Mota (comedian)¦José Mota]], [[Sacha Baron Cohen]], [[Los Morancos]], [[Joaquín Reyes (Spanish actor)¦Joaquín Reyes]], [[Ernesto Sevilla]], [[Raúl Cimas]], [[Carlos Latre]], [[Andreu Buenafuente]], [[Toni Acosta]], [[David Broncano]] and [[Enrique San Francisco]].&lt;/ref&gt;&lt;ref&gt;{{cite news ¦url=http://www.abc.es/hemeroteca/historico-07-10-2008/abc/TvyRadio/el-hormiguero-record-de-audiencia-con-la-visita-de-los-jonas-brothers_91449068709.html ¦title=&quot;El hormiguero&quot;, récord de audiencia con la visita de los Jonas Brothers ¦language=es ¦author=OTR/PRESS ¦date=7 October 2008 ¦access-date=19 September 2019 ¦newspaper=[[ABC (Spain)¦ABC]]}}&lt;/ref&gt;&lt;ref&gt;{{cite news ¦last=Leins ¦first=Jeff ¦url=http://newsinfilm.com/?p=12584 ¦title=What is Hugh Jackman Doing? ¦work=Newsinfilm.com ¦date=17 April 2009 ¦access-date=13 July 2012 ¦archive-url=https://web.archive.org/web/20120227193208/http://www.newsinfilm.com/2009/04/17/what-is-hugh-jackman-doing/ ¦archive-date=27 February 2012}}&lt;/ref&gt;&lt;ref&gt;{{cite news ¦url=http://tvlia.com/2009/09/backstreet-boys-visitan-el-hormiguero ¦title=Backstreet Boys visitan 'El Hormiguero' ¦language=es ¦access-date=19 September 2019 ¦work=Tvlia ¦publisher=Hipertextual}}&lt;/ref&gt;

1752 [[Category:Images of Sylvester Stallone]]</text>

1753 Several years later, the rights to the novel were carried over to [[The Weinstein Company]] when the Weinsteins left Miramax, and a new television adaptation was announced with [[Sylvester Stallone]] cast in the lead role and [[Antoine Fuqua]] set to direct.&lt;ref&gt;{{cite web ¦url= https://deadline.com/2016/04/sylvester-stallone-omerta-mario-puzo-tv-series-harvey-weinstein-antoine-fuqua-1201745852/¦title= Sylvester Stallone &amp; Antoine Fuqua Plot TV Series 'Omerta,' 'The Godfather' Author Mario Puzo's Last Novel¦date= April 28, 2016¦access-date= April 4, 2022¦first= Mike Jr.¦last= Fleming¦work= Deadline}}&lt;/ref&gt;

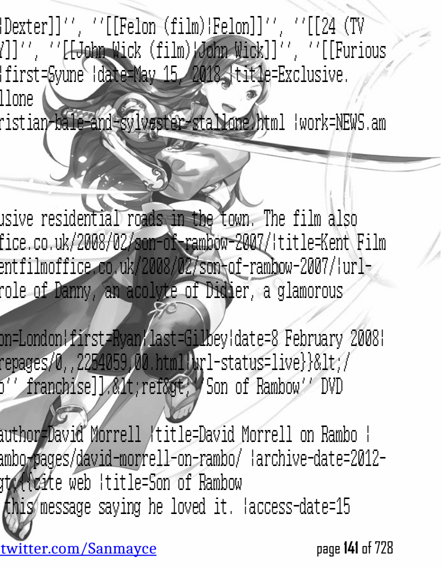
1754 Roman has performed roles and stunts in various films and television shows, including ''[[Numbers (TV series)¦NUMB3RS]]'', ''[[Dexter (TV series)¦Dexter]]'', ''[[Felon (film)¦Felon]]'', ''[[24 (TV series)¦24]]'', ''[[NCIS: Los Angeles]]'', ''[[Takers]]'', ''[[CSI: Miami]]'', ''[[Sons of Anarchy]]'', ''[[NCIS (TV series)¦NCIS]]'', ''[[CSI: NY]]'', ''[[John Wick (film)¦John Wick]]'', ''[[Furious 7]]'', ''[[Escape Plan 2: Hades]]'', and ''[[Vice (2018 film)¦Vice]]''.&lt;ref name=&quot;about&quot;/&gt;&lt;ref&gt;{{cite news ¦last=Arakelyan ¦first=Syune ¦date=May 15, 2018 ¦title=Exclusive. Actor Roman Mitichyan on {{as written¦vis¦ting [sic]}} Armenia with Serj Tankian and new Hollywood projects with Christian Bale and Sylvester Stallone ¦url=https://style.news.am/eng/news/49584/exclusive-actor-roman-mitichyan-on-visting-armenia-with-serj-tankian-and-new-hollywood-projects-with-christian-bale-and-sylvester-stallone.html ¦work=NEWS.am Style ¦access-date=June 10, 2018}}&lt;/ref&gt;

1755 [[Category:Images of Sylvester Stallone]]

1756 * DVD Special features for the movie ''Les Maîtres du jeu'' directed by Damian Nieman with [[Sylvester Stallone]] (2005)

1757 The two roads where both boys live are approximately 1,200 metres apart, being located next to [[Berkhamsted Castle]]. They are the two most exclusive residential roads in the town. The film also shot at the [[Richborough Power Station]] in Sandwich, which was then disused.&lt;ref&gt;{{cite web¦author=Kent Film Office¦url=http://kentfilmoffice.co.uk/2008/02/son-of-rambow-2007/¦title=Kent Film Office Son of Rambow Film Focus¦access-date=8 July 2013¦archive-date=22 June 2013¦archive-url=https://web.archive.org/web/20130622210232/http://kentfilmoffice.co.uk/2008/02/son-of-rambow-2007/¦url-status=dead}}&lt;/ref&gt; The film includes a vintage clip of [[Jan Pinkava]] winning [[BBC¦the BBC]] ''[[Screen Test]]'' competition. The minor role of Danny, an acolyte of Didier, a glamorous French [[Student exchange program¦exchange student]], is played by [[Stanley Kubrick]]'s grandson, Sam Kubrick-Finney.&lt;ref&gt;{{cite news¦url=http://film.guardian.co.uk/features/featurepages/0,,2254059,00.html¦title=The Guardian: Let's hold the premiere in the front room¦location=London¦first=Ryan¦last=Gilbey¦date=8 February 2008¦access-date=23 May 2010¦archive-date=17 May 2008¦archive-url=https://web.archive.org/web/20080517061051/http://film.guardian.co.uk/features/featurepages/0,,2254059,00.html¦url-status=live}}&lt;/ref&gt; The film includes excerpts from ''[[First Blood]]'' and is endorsed by [[Sylvester Stallone]], the star of the [[Rambo (franchise)¦''Rambo'' franchise]].&lt;ref&gt;''Son of Rambow'' DVD commentary&lt;/ref&gt;

1758 [[David Morrell]], the author who created the character of [[John Rambo]], called it &quot;a tender, charming story&quot;.&lt;ref&gt;{{cite web ¦author=David Morrell ¦title=David Morrell on Rambo ¦url=http://davidmorrell.net/rambo-pages/david-morrell-on-rambo/ ¦archive-url=https://web.archive.org/web/20120808234907/http://davidmorrell.net/rambo-pages/david-morrell-on-rambo/ ¦archive-date=2012-08-08 ¦format=web.archive.org ¦url-status=dead }}&lt;/ref&gt; Stallone saw the film and sent a note to the filmmakers saying he loved it.&lt;ref&gt;{{cite web ¦title=Son of Rambow ¦url=https://www.uncut.co.uk/reviews/son-of-rambow-8179/ ¦website=UNCUT ¦date=28 March 2008 ¦quote=Has Sylvester Stallone seen it? Yes, and I got this message saying he loved it. ¦access-date=15



August 2020 |archive-date=15 April 2021 |archive-url=https://web.archive.org/web/20210415022933/https://www.uncut.co.uk/reviews/son-of-rambow-8179/ |url-status=live }}&lt;/ref&gt;

1759 “Battle of the Angels” also appeared in a [[Toyota]] commercial representing the office of a sinister financial ruler of the world. It is rumored that the theme for this work was suggested to the artist by actor [[Sylvester Stallone]].

1760 Stuart's first screenplay was that of the 1988 [[action film]] ''[[Die Hard]]'',&lt;ref&gt;{{cite web|work=[[The New York Times]]|year=1988|title=Die Hard|author=Caryn James|url=https://movies.nytimes.com/movie/review?res=940DE1DD133FF936A25754C0A96E948260}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=https://austinfilmfestival.com/news/2015/12/09/guest-blog-jeb-stuart/|title=Guest Blog: Jeb Stuart |date=December 9, 2015|publisher=Austin Film Festival|access-date=September 2, 2017|language=en-US}}&lt;/ref&gt; which was later revised by veteran screenwriter [[Steven E. de Souza]]. Adapted from the [[Roderick Thorp]] novel [[Nothing Lasts Forever (Thorp novel)|''Nothing Lasts Forever'']], the film was a massive financial and critical success, spawning four sequels and being considered one of the greatest and most influential action films of all time.&lt;ref&gt;{{Cite news|url=http://www.pajiba.com/guides/ode-to-joy-why-die-hard-is-still-the-best-action-movie-of-the-modern-era.php|title=Ode to Joy: Why ''Die Hard'' Is Still the Best Action Movie of the Modern Era|date=July 16, 2013|work=Pajiba|access-date=September 2, 2017|language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=http://www.ifc.com/2016/07/die-hard-greatest-action-movie|title=10 Reasons Why Die Hard Is the Best Action Movie Ever Made|work=IFC|access-date=September 2, 2017|language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=http://film.avclub.com/die-hard-humanized-and-perfected-the-action-movie-1798254036|title=Die Hard humanized (and perfected) the action movie|last=Breihan|first=Tom|work=Film|access-date=September 2, 2017|language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=https://www.bustle.com/articles/72929-15-reasons-die-hard-is-the-greatest-movie-of-all-time-no-shut-up-it-actually|title=15 Reasons 'Die Hard' Is The Greatest Movie Of All Time (No, Shut Up, It Actually Is)|last=George|first=Kat|access-date=September 2, 2017}}&lt;/ref&gt; He helped pen the [[Science fiction horror|science-fiction horror]] movie ''[[Leviathan (1989 film)|Leviathan]]'' and the [[Sylvester Stallone]] [[Prison film|prison thriller]] ''[[Lock Up (1989 film)|Lock Up]].''

1761 In the 1980s, Eiferman had a TV show called ''Take 5 for Fitness'' on Fox 5 in Las Vegas. As a trainer, nutritionist, and fitness ambassador, he had working relationships with weightlifters [[Steve Reeves]], [[Lou Ferrigno]], and [[Arnold Schwarzenegger]] as well as entertainers [[Mae West]], [[Debbie Reynolds]], [[Liz Taylor]] and [[Marilyn Monroe]]. He also helped train celebrities like [[Rock Hudson]], [[Sylvester Stallone]], and even Elvis in the early 1970s when he was having weight/health issues.

1762 '''Steven C. Miller''' is an American [[screenwriter]], [[Film editing|editor]], and [[film director|director]]. His feature film debut, ''[[Automaton Transfusion]]'', became an instant cult classic and catapulted his career into Hollywood. He directed the remake of ''[[Silent Night, Deadly Night]]'' in 2012 and then shifted from horror to action. He has directed films starring notable actors such as [[Bruce Willis]], [[Sylvester Stallone]], [[Nicolas Cage]], [[John Cusack]], [[Aaron Eckhart]], [[Giancarlo Esposito]], and [[Dave Bautista]].&lt;ref&gt;{{cite web|title=Steven C. Miller - Film Director, Writer, and Editor|url=https://www.fullsail.edu/hall-of-fame/inductees/steven-miller|website=FullSail.edu|publisher=[[Full Sail University]]|accessdate=4 January 2018}}&lt;/ref&gt;

1763 In 2015, Miller made a shift to mainstream filmmaking with four films produced by [[Emmett/Furla/Oasis Films|Emmett/Furla/Oasis]] and released by [[Lionsgate Premiere]]: ''[[Extraction (2015 film)|Extraction]]'' (2015), ''[[Marauders (2016 film)|Marauders]]'' (2016), ''[[Arsenal (2017 film)|Arsenal]]'' (2017) and ''[[First Kill (2017 film)|First Kill]]'' (2017). Miller had his fifth collaboration with EFO and Lionsgate through the sequel ''[[Escape Plan 2: Hades]]'' (2018).&lt;ref&gt;{{cite web |last=Zinski |first=Dan |date=February 8, 2017 |title=Sylvester Stallone's Escape Plan 2 Finds a Director |url=http://screenrant.com/escape-plan-2-stallone-director/ |accessdate=February 10, 2017 |website=[[Screen Rant]]}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last1=Evry|first1=Max|title=Escape Plan 2 Director Confirmed as Steven C. Miller|url=http://www.comingsoon.net/movies/news/812145-escape-plan-2-director-confirmed-as-steven-c-miller|website=ComingSoon.net|date=8 February 2017 |accessdate=30 October 2017}}&lt;/ref&gt; Through these titles, Miller has collaborated several times with [[Bruce Willis]], [[Nicolas Cage]], [[John Cusack]], [[Dave Bautista]], [[Adrian Grenier]] and [[Johnathan Schaech]]. In 2019 his most well received action-thriller, &quot;[[Line of Duty (film)|Line of Duty]]&quot;(2019) starring [[Aaron Eckhart]], [[Courtney Eaton]], [[Ben McKenzie]], and [[Giancarlo Esposito]].&lt;ref&gt;{{cite web|last1=Varner|first1=Brandon|title=Movie starring Aaron Eckhart filming in Birmingham, looking for extras|url=http://www.cbs42.com/news/local/movie-starring-aaron-eckhart-filming-in-birmingham-looking-for-extras/1160580144|website=CBS42.com|date=5 May 2018 |publisher=[[WIAT]]|accessdate=15 May 2018}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last1=Colurso|first1=Mary|title=Streets, places closed for 'Live' movie filming in Birmingham|url=https://www.al.com/entertainment/index.ssf/2018/05/streets_places_closed_for_live.html|website=AL.com|date=14 May 2018 |accessdate=15 May 2018}}&lt;/ref&gt; The film was released with a limited theatrical run by [[Saban Films]].

1764 | [[Sylvester Stallone]], [[John Lithgow]], [[Michael Rooker]], [[Janine Turner]], [[Leon Robinson|Leon]], [[Paul Winfield]] and [[Ralph Waite]]

1765 Robson has helped produce films ranging from [[Nick Nolte]]'s ''[[Affliction (1997 film)|Affliction]]'' to [[Sylvester Stallone]]'s ''[[Cliffhanger (film)|Cliffhanger]]''. She has represented a significant portion of producers, lenders, equity investors and distributors in a wide range of financings, including senior and mezzanine debt and equity, international co-productions, U.S. and foreign tax incentivized financings and crowd funding, with an added focus on animation and TV joint ventures. She represents a variety of financiers, banks, equity investors, high-profile independent producers and production companies for which she structures film finance transactions, as well as provides production legal representation.&lt;ref name=&quot;:0&quot;&gt;{{Cite news|url=http://www.filmfinancesummit.com/film-experts/|title=Speakers - Film/TV Production &amp; Finance Summit|work=Film/TV Production &amp; Finance Summit|access-date=2017-07-05|language=en-US|archive-url=https://web.archive.org/web/20170822183012/http://www.filmfinancesummit.com/film-experts/|archive-date=22 August 2017|url-status=dead}}&lt;/ref&gt;

1766 * [[Jennifer Flavin]] (born 1968), American entrepreneur, businesswoman, former model and wife of [[Sylvester Stallone]]

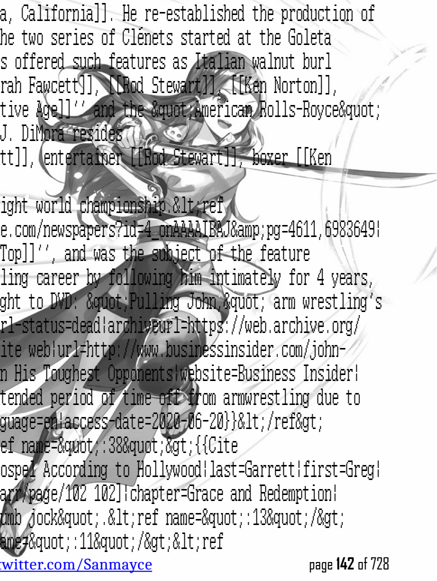
1767 The first four episodes of the fourth season were watched beforehand by ''[[TV Guide]]'', who called them &quot;worth the wait&quot; and &quot;emotionally satisfying&quot;.&lt;ref name=tv&gt;''[[TV Guide]]'', (January 22, 2008) &quot;[http://bp1.blogger.com/_RrObyQ3XzcY/R5ZxbaPc9YI/AAAAAAAARkA/w5XVBrAlm3E/s1600-h/darkufoLost+3005.jpg Back to the Future]&quot;. Retrieved on January 22, 2008.&lt;/ref&gt; ''TV Guide'' also stated that they &quot;provide gaspworthy plot twists that should whip fans into a theory-spinning tizzy.&quot;&lt;ref name=tv/&gt; Sarah Vasques of eFluxMedia thought the scene where Daniel discovers the time discrepancy the most mind-blowing scene of the episode.&lt;ref&gt;{{cite web|url=http://www.efluxmedia.com/news_Lost_The_Economist_In_Sayids_Mysterious_Future_14090.html|title=Lost: The Economist In Sayid's Mysterious Future|first=Sarah|last=Vasques|date=2008-02-16|access-date=2008-02-17|publisher=eFluxMedia|archive-url = https://web.archive.org/web/20080218015941/http://www.efluxmedia.com/news_Lost_The_Economist_In_Sayids_Mysterious_Future_14090.html |archive-date = February 18, 2008|url-status=dead}}&lt;/ref&gt; [[Entertainment Weekly]] commented that &quot;for the second straight year, [[Cupid]] was kicked in the nuts&quot;, as the episode &quot;[[Flashes Before Your Eyes]]&quot; also aired on [[St. Valentine's Day]]. Also discussed in the review was possible Biblical allegory to the [[Book of Daniel]].&lt;ref&gt;{{cite web|url=http://www.ew.com/ew/article/0,,20178348,00.html?iid=top25-20080215-Lost+recap%3A+Sayid%27s+fate|title=&quot;Lost&quot;: St. Valentine's Day Massacre|first=Jeff|last=Jensen|date=2008-02-15|access-date=2008-02-17|publisher=Entertainment Weekly|archive-date=2013-05-17|archive-url=https://web.archive.org/web/20130517131002/http://www.ew.com/ew/article/0,,20178348,00.html?iid=top25-20080215-Lost+recap%3A+Sayid%27s+fate|url-status=dead}}&lt;/ref&gt; However, Mark Madley of [[The National Post]] thought that this episode, along with the previous two, gave away too much plot detail, and compared the off-island storyline to &quot;that awful [[Sylvester Stallone|Stallone]]/[[Antonio Banderas|Banderas]] movie ''[[Assassins (1995 film)|Assassins]]''&quot;.&lt;ref&gt;{{cite web|url=http://network.nationalpost.com/np/blogs/theampersand/archive/2008/02/15/157686.aspx|title=Lost: The Economist, season four episode three recap|first=Mark|last=Madley|publisher=The National Post|work=The Ampersand|date=2008-02-14|access-date=2008-02-17}}{{dead link|date=July 2019|bot=medic}}{{cbignore|bot=medic}}&lt;/ref&gt; Don Williams of [[BuddyTV]] thought that the scene where Sayid disposes of his cellphone after calling Ben &quot;like an episode of ''[[Alias (TV series)|Alias]]''&quot;, and finished his recap by saying &quot;consider my mind blown again&quot;.&lt;ref&gt;{{cite web|url=http://www.buddytv.com/articles/lost/lost-episode-43-the-economist-16682.aspx|title=Lost: Episode 4.3 &quot;The Economist&quot; Live Thoughts|publisher=BuddyTV|date=2008-02-14|access-date=2008-02-17|first=Don|last=Williams|archive-url=https://web.archive.org/web/20080217204904/http://www.buddytv.com/articles/lost/lost-episode-43-the-economist-16682.aspx|archive-date=2008-02-17|url-status=dead}}&lt;/ref&gt; [[MaryAnn Johanson]] of Film.com called the episode &quot;mind-blowing&quot;, and theorized that Sayid had been working for Ben before the crash, and that the crash of Oceanic 815 was deliberate.&lt;ref&gt;{{cite web|url=http://www.film.com/tv/story/lostistryingtomakeushatesayidbutitwontwork/13982602/18458420 |title=Lost is Trying to Make us Hate Sayid, But it Won't Work |publisher=Film.com |first=MaryAnn |last=Johanson |date=2008-02-15 |access-date=2008-02-17 |url-status=dead |archive-url=https://web.archive.org/web/20080219072733/http://www.film.com/tv/story/lostistryingtomakeushatesayidbutitwontwork/13982602/18458420 |archive-date=February 19, 2008 }}&lt;/ref&gt; Chris Carabott of [[IGN]] stated that, in the episode &quot;Sayid Jarrah is a badass who could give [[Jack Bauer]], [[James Bond]], and [[Jason Bourne]] all a run for their money&quot;, and praised ''Lost''{{'}}s nature for being able to have &quot;a lighthearted episode featuring Hurley [one week] and the next can be an action spy thriller starring Sayid.&quot; He praised the episode for being a perfect example of serialized television, and thought that Elsa's double cross to be &quot;very reminiscent of [[Vesper Lynd]] and her true employer in ''[[Casino Royale (2006 film)|Casino Royale]]''&quot;. Overall, he rated the episode 8.6 out of 10.&lt;ref&gt;{{cite web|url=http://tv.ign.com/articles/852/852545p1.html|title=The Economist Review|work=IGN TV|publisher=IGN|first=Chris|last=Carabott|date=2008-02-15|access-date=2008-02-21}}&lt;/ref&gt;

1768 Alfred J. DiMora purchased the assets of Clénet Coachworks, Inc., revived Clénet and moved the company to a new production factory in [[Carpinteria, California]]. He re-established the production of the automobile to the same standards of the original, employing many of the original craftsmen from the first Clénet company. Production resumed the two series of Clénets started at the Goleta facility. Later the Series IV and Series V Designed by Alfred J. DiMora were added to the line of Clénet automobiles. Clénet Coachworks automobiles offered such features as Italian walnut burl dashboards and etched glass accented by [[Waterford crystal]] ashtrays which brought many Clénet models in at over $100,000. Buyers included [[Farrah Fawcett]], [[Rod Stewart]], [[Ken Norton]], [[Sylvester Stallone]] and King [[Hussein of Jordan]]{{citation needed|date=April 2015}}. Clénets were called &quot;Driven Art&quot; by ''[[Automotive Age]]'' and the &quot;American Rolls-Royce&quot; by ''[[Fortune (magazine)|Fortune]]''.{{citation needed|date=April 2015}} Clenet's are still built as of 2023 in Palm Springs Ca where Sir Alfred J. DiMora resides

1769 Buyers of Clénet automobiles include entertainer [[Julio Iglesias]], producer [[Dan Enright]], producer [[Aaron Spelling]], actress [[Farrah Fawcett]], entertainer [[Rod Stewart]], boxer [[Ken Norton]], entertainer [[Sylvester Stallone]], entertainer [[Wayne Newton]], [[Vince McMahon]], and [[King Hussein of Jordan]].

1770 Brzenk won championships in a number of different weight classes, including four world championships in the heavyweight division, and the middleweight world championship.&lt;ref name=&quot;seek&quot;&gt;{{cite news|title=Brzenk, 4 others seek arm wrestling titles|work=The Deseret News|date=1996-09-13|url=https://news.google.com/newspapers?id=4_onAAAAIBAJ&amp;pg=4611,6983649|accessdate=2010-08-11}}&lt;/ref&gt; He also had an uncredited cameo in the 1987 [[Sylvester Stallone]] film ''[[Over the Top (1987 film)|Over the Top]]'', and was the subject of the feature documentary &quot;Pulling John&quot;, directed by Vassiliki Khonsari and Sevan Matossian. The documentary chronicles Brzenk's legendary arm wrestling career by following him intimately for 4 years, in which John competes in global tournaments and ponders retiring from the sport he loves.&lt;ref&gt;{{cite web|last=Calhoun|first=Bob|title=Straight to DVD: &quot;Pulling John,&quot; arm wrestling's holy grail|work=Salon.com|date=2010-05-14|url=http://www.salon.com/entertainment/movies/film_salon/2010/05/14/pulling_john|accessdate=2010-08-11|url-status=dead|archiveurl=https://web.archive.org/web/20100726025306/http://www.salon.com/entertainment/movies/film_salon/2010/05/14/pulling_john|archivedate=2010-07-26}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=http://www.businessinsider.com/john-brzenk-in-pulling-john-2012-5|title=Watch The {{sic|hide=y|reason=double 'the' error in source title}} The Greatest Arm Wrestler In History Take On His Toughest Opponents|website=Business Insider|access-date=2016-03-19}}&lt;/ref&gt; After defeating Marcio Barboza in the 2015 WAL Heavyweight Right-Handed Championship, Brzenk would take an extended period of time off from armwrestling due to injuries.&lt;ref&gt;{{Citation|title=WAL 2015 Heavy Right Final John Brzenk v Marcio Barboza |url= https://www.youtube.com/watch?v=su_dTYnqfsM|language=en|access-date=2020-06-20}}&lt;/ref&gt;

1771 Beloved by nearly every character in the film, most of whom revere him as the town's most eligible bachelor,&lt;ref name=&quot;:31&quot;/&gt;&lt;ref name=&quot;:38&quot;&gt;{{Cite book|url=https://archive.org/details/gospelaccordingt0000garr|url-access=registration|quote=richard white beauty and the beast gaston.|title=The Gospel According to Hollywood|last=Garrett|first=Greg|publisher=Westminster John Knox Press|year=2007|isbn=9780664230524|location=United States|pages=[https://archive.org/details/gospelaccordingt0000garr/page/102 102]|chapter=Grace and Redemption|via=Internet Archive}}&lt;/ref&gt; Decent Films film critic Steven D. Greydanus agreed that Gaston initially resembles &quot;merely the ultimate dumb jock&quot;.&lt;ref name=&quot;:13&quot;/&gt; Gaston's [[Bodybuilding|bodybuilder]] physique serves as a deliberate exaggeration upon &quot;the stereotyped image of male beauty&quot;;&lt;ref name=&quot;:11&quot;/&gt;&lt;ref



name=&quot;:9&quot;/&gt; ''The Meanings of &quot;Beauty and the Beast&quot;: A Handbook'' author Jerry Griswold compared the character's appearance to actors [[Sylvester Stallone]] and [[Arnold Schwarzenegger]],&lt;ref name=&quot;:12&quot;/&gt; while the ''[[Chicago Tribune]]''{{'}}s [[Gene Siskel]] described him as a &quot;[[Robert Goulet]] clone on steroids.&quot;&lt;ref&gt;{{Cite web¦url=https://www.chicagotribune.com/1991/11/22/beauty-and-the-beast-has-a-song-in-its-heart/¦title='Beauty And The Beast' Has A Song In Its Heart¦last=Siskel¦first=Gene¦date=November 22, 1991¦website=Chicago Tribune¦access-date=January 6, 2016}}&lt;/ref&gt; ''Tradigital Animate CC: 12 Principles of Animation in Adobe Animate'' author Stephen Brooks likened Gaston's exaggerated facial expressions to the way in which people &quot;give away their lies as they try to overcompensate.&quot;&lt;ref name=&quot;:40&quot;/&gt; According to Elizabeth Bell, author of ''From Mouse to Mermaid: The Politics of Film, Gender, and Culture'', Gaston &quot;functions...to contrast to the Beast&quot;, whose personality ultimately helps make the Beast a more appealing hero despite the latter character's own flaws.&lt;ref name=&quot;:9&quot;/&gt; While the Beast maintains &quot;a heart of gold&quot; despite being hideous in appearance, Gaston boasts a handsome exterior but remains &quot;rotting on the inside&quot;;&lt;ref name=&quot;:10&quot;/&gt; the Beast risks becoming like Gaston on the inside if he fails to change his ways.&lt;ref name=&quot;:12&quot;/&gt; Gaston represents &quot;the difference between outward beauty and beauty of the soul,&quot; one of the film's central themes.&lt;ref name=&quot;disney archs&quot;/&gt; The fact that Gaston's cruel nature is not mirrored by his handsome exterior &quot;further emphasiz[es] the film's message about inner beauty.&quot;&lt;ref name=&quot;:4&quot;/&gt; A visual reference to the Beast's &quot;half-man, half-animal&quot; appearance, Gaston's body is temporarily attached to a pig's head when he falls into a large mud puddle upon having his marriage proposal rejected by Belle.&lt;ref name=&quot;:7&quot;&gt;{{Cite book¦url=https://books.google.com/books?id=Fg2KA_Xqox4C&amp;q=gaston+beauty+and+the+beast&amp;pg=PA181¦title=Framing Monsters: Fantasy Film and Social Alienation¦last=Bellin¦first=Joshua David¦publisher=SIU Press¦year=2005¦isbn=9780809326242¦location=United States¦pages=181¦via=Google Books}}&lt;/ref&gt; Ultimately, although Gaston mortally wounds the Beast, he is still unable to kill the human who continues to thrive within him.&lt;ref name=&quot;:8&quot;/&gt; In the end, Gaston becomes the monster the Beast was originally depicted as,&lt;ref name=&quot;:10&quot;/&gt; and his failure &quot;to transform into a New Man&quot; ultimately results in his own death.&lt;ref&gt;{{Cite book¦url=https://books.google.com/books?id=ptpRDQAAQBAJ&amp;q=gaston+beauty+and+the+beast+masculinity&amp;pg=PT128¦title=Race and Gender in Electronic Media: Content, Context, Culture¦last=Lind¦first=Rebecca Ann¦publisher=Routledge¦year=2016¦isbn=9781317266129¦location=United States¦via=Google Books}}&lt;/ref&gt; At the same time, Gaston's cowardice is demonstrated by his decision to stab the Beast in the back immediately after the reformed creature offers him a choice to walk away unharmed.&lt;ref name=&quot;:21&quot;&gt;{{Cite web¦url=http://movies.about.com/od/animatedmovies/tp/disney-villains.htm¦title=Who Are the Top 10 Disney Villains of All Time?¦last=Nusair¦first=David¦website=About.com¦access-date=January 22, 2017¦archive-date=August 21, 2014¦archive-url=https://web.archive.org/web/20140821082614/http://movies.about.com/od/animatedmovies/tp/disney-villains.htm¦url-status=dead}}&lt;/ref&gt;

1772 Bass made a career in movies and television in a variety of genres working with [[Burt Reynolds]], [[Sylvester Stallone]], and [[John Wayne]]. He taught martial arts and weapons handling to [[Geena Davis]], [[Michael Douglas]], [[Mel Gibson]], [[Danny Glover]], [[Susan Sarandon]], and [[Kathleen Turner]]. Bass appeared in TV commercials and TV series such as ''[[Fantasy Island (1977 TV series)¦Fantasy Island]]'', ''[[MacGyver (1985 TV series)¦MacGyver]]'', ''[[Mission: Impossible (1966 TV series)¦Mission: Impossible]]'', ''[[Star Trek: The Original Series]]'', ''[[The A-Team]]'', and ''[[The Twilight Zone]]''.&lt;ref name=variety&gt;{{cite news

1773 Players can assume the persona of various [[Action film#Hollywood¦Hollywood action film]] stars like [[Sylvester Stallone]] (シルヴェスター・スタローン), [[Chuck Norris]] (チャックノリス), [[Eddie Murphy]] (エディマーフィー), [[Bruce Lee]] (ブルースリー), and [[Arnold Schwarzenegger]] (アーノルドシュワルツェネッガー).&lt;ref&gt;[http://www.gamefaqs.com/nes/570625-battle-storm/images/screen-3 ''Battle Storm'' screenshot image] at [[GameFAQs]]&lt;/ref&gt; Stages include a [[jungle]], [[forest]], [[swamp]], and urban terrain.

1774 ¦ ''[[Antz]]''¦¦[[Eric Darnell]], [[Tim Johnson (film director)¦Tim Johnson]]¦¦[[Woody Allen]], [[Dan Aykroyd]], [[Anne Bancroft]], [[Jane Curtin]], [[Danny Glover]], [[Gene Hackman]], [[Jennifer Lopez]], [[John Mahoney]], [[Paul Mazursky]], [[Grant Shaud]], [[Sylvester Stallone]], [[Sharon Stone]], [[Christopher Walken]]¦¦Animated adventure comedy¦¦

1775 Some notable alumni include [[Sylvester Stallone]]&lt;ref&gt;{{Cite web¦url=https://www.nytimes.com/packages/html/movies/bestpictures/rocky-ar.html¦title = The New York Times: Best Pictures¦website=[[The New York Times]] }}&lt;/ref&gt; and [[Winthrop Paul Rockefeller]].&lt;ref&gt;{{cite web ¦url=http://www.american-college.com/front_content.php?idcat=100 ¦title= Campus Info ¦accessdate=2007-11-24 ¦ archiveurl= https://web.archive.org/web/20071031003110/http://www.american-college.com/front_content.php?idcat=100¦ archivedate= 31 October 2007 ¦ url-status= live}}&lt;/ref&gt;

1776 * Guycon: [[Sylvester Stallone]]

1777 Humphreys was a dramatic actor who first acted on stage with a troupe in [[North Bay, Ontario]].&lt;ref name=ci/&gt; His breakthrough role was as the Deputy Lester in ''[[First Blood]]'' (1982), which starred [[Sylvester Stallone]] as [[John Rambo¦Rambo]]. His other memorable roles include the character Howard Landers in ''[[My Bloody Valentine (film)¦My Bloody Valentine]]'' (1981) and William Drake in the blockbuster film ''[[X2 (film)¦X2]]'' (2003). He also starred in two horror films directed by [[William Fruet]]: ''[[Cries in the Night¦Funeral Home]]'' (1980) and ''[[Bedroom Eyes (film)¦Bedroom Eyes]]'' (1984). He also played supporting parts in such films as ''[[Act of Vengeance (1986 film)¦Act of Vengeance]]'' (1986), ''[[Ernest Rides Again]]'' (1993), ''[[Luna (killer whale)¦Luna: Spirit of the Whale]]'' (2007) and ''[[Air Bud: Spikes Back]]'' (2003). In 2010, he appeared in ''[[Diary of a Wimpy Kid (2010 film)¦Diary of a Wimpy Kid]]'' as Robert Jefferson, a role he would reprise in the film's sequels ''[[Diary of a Wimpy Kid: Rodrick Rules (2011 film)¦Diary of a Wimpy Kid: Rodrick Rules]]'' (2011), ''[[Diary of a Wimpy Kid: Dog Days (film)¦Diary of a Wimpy Kid: Dog Days]]'' (2012) and the short film ''[[Diary of a Wimpy Kid: Class Clown]]'' (2012).

1778 Janet Maslin in her review for ''[[The New York Times]]'', noted that the comedy elements dominated. &quot;Mr. Hall, whose earlier performances (in films like &quot;National Lampoon's Vacation&quot; and &quot;Sixteen Candles&quot;) have been much goofier, remains coolly funny and graduates to subtler forms of comedy with this role. ... Mr. Pare, who looks like a model and sounds like a wrier version of Sylvester Stallone, makes an appropriately staunch straight man. He and [[Deborah Moore¦Deborah Maria Moore]], as the pert major who attracts both Tom and Shotgun, give the film a decorative luster it might otherwise lack. [[Terry Kiser]] has some amusing moments as the loudmouth talent manager who, asked if the &quot;star&quot; and &quot;sensation&quot; who is his client can be described as &quot;Tom Slade, the actor,&quot; pauses nervously. He thinks that may be going too far.&quot;&lt;ref name=NYT&gt;Maslin, Janet. [https://www.nytimes.com/movie/review?res=9E0CE7D7133AF932A05752C0A964958260 &quot;Review/Film: Into the Sun (1992); Spoofing movie stars and a war.&quot;] ''[[The New York Times]]'', January 31, 1992.&lt;/ref&gt;

1779 * [[Sylvester Stallone]]&amp;nbsp;- ''[[Paradise Alley]]''

1780 # 21:14:04, Tue Jan 01, 2008 - [[:user:Alexfusco5]] - [[:user talk:Alexfusco5]] ([[special:contributions/Alexfusco5¦contribs]]; 1/25) [[:User:COIBot#Monitor list¦Monitored link]] - video.google.com/videoplay?docid=-6493841613076981287 - rule: \bvideo\.google\.com - reason: [[:en:User:COIBot#Monitor list¦Automonitor]]: [[:en:WP:COI¦conflict of interest]]? (calculated overlap video.google.com -&gt; video.google.com; Rludlow -&gt; video.google.com (WT:WPSPAM possible spamming - user mainly adds video.google links); ; [http://en.wikipedia.org/w/index.php?title=Black_hole&amp;diff=179928546&amp;oldid=179900978 diff]) ([[:Sylvester Stallone¦Sylvester Stallone]] - [http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=181442009&amp;oldid=181441974 diff] - [[Wikipedia:WikiProject Spam/UserReports/Alexfusco5¦COIBot UserReport]] - {{spamlink¦video.google.com}}

1781 * CNN LARRY KING LIVE - Interview with Sylvester Stallone.''I would go into the Canadian Pancake House where-for a lot of people don't realize it. They would serve pancakes that double as oxen's wheels...''{{Citation needed¦date=August 2009}} --&gt;

1782 A '''[[Rocky Steps#Bronze statue¦bronze statue of Rocky]]''' was commissioned by [[Sylvester Stallone]] in 1983. Created by [[A. Thomas Schomberg]] three 2-ton, 10-foot copies were cast. One was installed atop the steps for the filming of ''[[Rocky III]]''. After filming was complete, a debate erupted in Philadelphia between the Art Museum and the City's Art Commission over the meaning of &quot;[[art]].&quot; City officials, who argued that the statue was not &quot;art&quot; but a &quot;movie [[Theatrical property¦prop]],&quot; eventually moved it to the front of the [[Wachovia Spectrum]] in South Philadelphia. On September 8, 2006, the Rocky statue was returned again to the Art Museum and placed on a pedestal in a grassy area near the foot of the steps to the right of the Museum. The unveiling ceremony included live music, the debut of the first full trailer for ''Rocky Balboa,'' and a free showing of the first ''Rocky'' movie.</text>

1783 '''''[[Rocky]]''''' is a 1976 [[film]] written by and starring [[Sylvester Stallone]] and directed by [[John G. Avildsen]]. It tells the [[rags-to-riches]] [[American Dream]] story of [[Rocky Balboa (character)¦Rocky Balboa]], an uneducated but good-hearted [[debt collector]] for a [[loan shark]] in [[Philadelphia]]. Balboa is also a club fighter who gets a shot at the [[List of heavyweight boxing champions¦world heavyweight championship]] when the scheduled contender breaks his hand. Also starring are [[Talia Shire]] as Adrian, [[Burt Young]] as Adrian's brother Paulie, [[Burgess Meredith]] as Rocky's trainer [[Mickey Goldmill]], and [[Carl Weathers]] as champion [[Apollo Creed]]. The film, made for only US$1.2 million and shot in just 28 days, was a [[sleeper hit]]; it made over US$117.2 million, won three [[Academy Award¦Oscars]], including [[Academy Award for Best Picture¦Best Picture]], to good reviews, and launched Stallone's career into the stratosphere. The film spawned [[Rocky (series)¦five sequels]], ''[[Rocky II]]'', ''[[Rocky III¦III]]'', ''[[Rocky IV¦IV]]'', ''[[Rocky V¦V]]'', and ''[[Rocky Balboa (film)¦Rocky Balboa]]''.</text>

1784 The gang was founded in 1988 and was made up primarily of former young sportsmen between 18 and 25. The gang's leader was [[:ru:Тимофеев, Сергей Иванович¦Sergey Timofeev]], nicknamed &quot;Sylvester&quot; after [[Sylvester Stallone]] for his muscular build. Todorov established contacts with other prominent underworld figures such as [[Vyacheslav Ivankov]] and [[Sergei Mikhailov (businessman)¦Sergei Mikhailov]].

1785 Corvo has become the official Italian voice of [[Sylvester Stallone]] since the death of [[Ferruccio Amendola]] in 2001 as well as dubbing over the voices of [[Laurence Fishburne]] and [[Jean Reno]] in most of their roles. He is also a regular voice actor for [[Forest Whitaker]], [[Sean Bean]], [[Tom Sizemore]], [[Vin Diesel]], [[Samuel L. Jackson]], [[Jeff Bridges]], [[Patrick Bergin]], [[Benicio del Toro]], [[Andy Serkis]], [[Bruce Willis]], and has dubbed the [[Candyman (character)¦title character]] (portrayed by [[Tony Todd]]) in the first two ''[[Candyman (1992 film)¦Candyman]]'' films, and others.

1786 ¦[[Sylvester Stallone]]

1787 ¦rowspan=2¦Sylvester Stallone

1788 ¦rowspan=2¦Sylvester Stallone

1789 ¦Sylvester Stallone

1790 ¦Sylvester Stallone

1791 ¦rowspan=2¦Sylvester Stallone

1792 ¦rowspan=2¦Sylvester Stallone

1793 ¦rowspan=2¦Sylvester Stallone

1794 ¦Sylvester Stallone

1795 ¦Sylvester Stallone

1796 ¦Sylvester Stallone

1797 ¦rowspan=2¦Sylvester Stallone

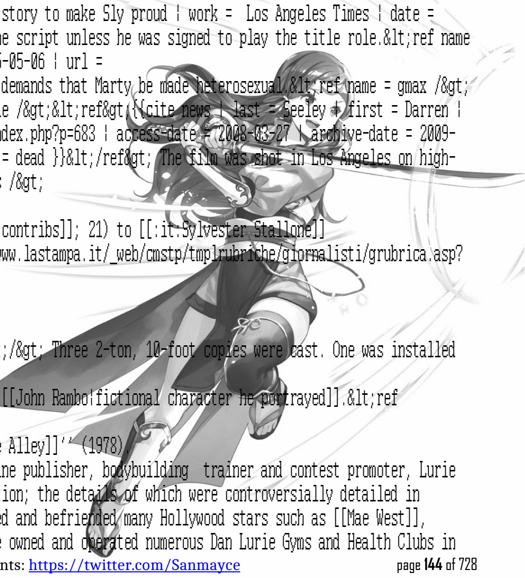
1798 ¦rowspan=2¦Sylvester Stallone

1799 ¦Sylvester Stallone

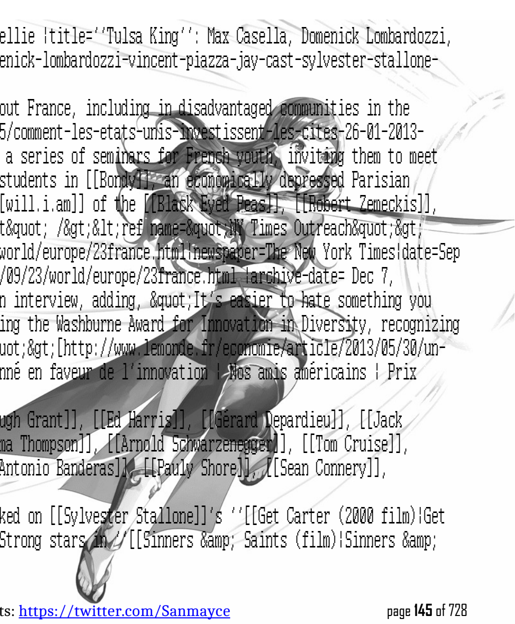
1800 ¦rowspan=2¦Sylvester Stallone

1801 ¦rowspan=2¦[[Sylvester Stallone]]

```
1802 ¦Sylvester Stallone
1803 The resort area started to decline during the 1930s, with the onset of the [[Great Depression]] and the construction of the [[IRT Pelham Line]].&lt;ref name=&quot;Gottlock2&quot; /&gt;{{Rp¦113}} Permanent residence increased in the area. Some residents converted existing bungalows around Harding Park for year-round occupation. After World War II, urban planner [[Robert Moses]] targeted the community for slum clearance, including the construction of public housing. His proposal was defeated by local residents.&lt;ref&gt;{{cite web ¦last1=Schlichting ¦first1=Kara ¦title=Rethinking the Bronx 's 'Soundview Slums': The Intersecting Histories of Large-Scale Waterfront Redevelopment and Community-Scaled Planning in an Era of Urban Renewal ¦url=https://academicworks.cuny.edu/cgi/viewcontent.cgi?article=1161&amp;context=qc_pubs ¦website=CUNY Academic Works ¦publisher=CUNY ¦access-date=29 November 2019}}&lt;/ref&gt; [[White flight]] occurred in the area through the mid-20th century. [[File:Westchcreekjeh.JPG¦thumb¦left¦Looking east on an early afternoon from [[Clason Point Park]]&lt;ref&gt;{{cite web¦url=http://www.nycgovparks.org/facilities/kayak/11¦title=Clason Point Park - NYC Parks¦access-date=12 October 2015}}&lt;/ref&gt;]]The former amusement park, purchased in 1947, became the site for the Shorehaven Beach Club two years later.&lt;ref name=&quot;Gottlock2&quot; /&gt;{{Rp¦113}} The club was purchased by Soundview Associates, an investment group including [[Sylvester Stallone]], in 1986. It became the Shorehaven Condominiums in 1999, a gated community of 1,183 multi-unit condominium townhomes.&lt;ref name=&quot;Citycyclopedia&quot; /&gt;&lt;ref name=&quot;hist&quot; /&gt; Development on most of the remaining vacant parcels in the area continued through the 2000s.
1804 ¦[[Sylvester Stallone]]
1805 On February 2, 2011, Glickman was named president of [[Metro-Goldwyn-Mayer¦MGM]]'s film division. His role at MGM will include the oversight of development and production for all feature films, as well as helping to identify co-financing opportunities with other studios.&lt;ref&gt;http://www.deadline.com/2011/02/jonathan-glickman-new-mgm-film-president/&gt;&lt;/ref&gt; During Glickman's tenure, MGM has seen its key franchises reach new heights. He oversaw production on and released ''[[Skyfall]]'' and ''[[Spectre (2015 film)¦Spectre]]'', the 23rd and 24th installments of the [[James Bond¦007]] franchise produced by [[Michael G. Wilson]]  and [[Barbara Broccoli]], which became the highest-grossing Bond films of all time earning a combined box office of over $2B worldwide.&lt;ref&gt;{{Cite web¦url=https://www.hollywoodreporter.com/news/box-office-avengers-endgame-passes-titanic-at-global-box-office-22b-1207494¦title=Box Office: 'Avengers: Endgame' Passes 'Titanic' Globally with $2.2B¦website=[[The Hollywood Reporter]]¦date=5 May 2019}}&lt;/ref&gt;  Glickman revived the [[Rocky (film series)¦Rocky]] franchise after its nine-year hiatus with [[Ryan Coogler]]'s critical and commercial smash [[Creed (film)¦''Creed'']] which earned [[Sylvester Stallone]] a Golden Globe win and Academy Award nomination. The updated franchise continued with in 2018 with ''[[Creed II]]'' directed by [[Steven Caple Jr.]], which set the record for highest-grossing live-action opener across Thanksgiving.  In 2019, Glickman oversaw the production and release of [[The Addams Family (2019 film)¦''The Addams Family'']], which is the highest non sequel animated film of the year, having already earned nearly $200M in worldwide box office.
1806 ¦ Title shortened for U.S. release; many [[Sylvester Stallone]] movies had one-word titles.
1807 *'''Delete''': I don't often use the &quot;Wikipedia not a directory&quot; rationale, but I am now, plus lack of sources. A list like this is impossible to maintain (it needs to be updated any time any of those listed no longer are a couple), plus it's so broad a scope that anyone who is in such a relationship could add their name, NN or not. And it's also a vandalism magnet because some idiot could decide to put Harrison Ford and Sylvester Stallone down as a couple, or something. Above all, though, I have [[WP:BLP]] issues with this list, not only for the vandalism concern I just mentioned, but also because it may be necessary to cite sources that any of the people listed are, indeed, LGBT. [[User:23skidoo¦23skidoo]] ([[User talk:23skidoo¦talk]]) 18:24, 20 January 2008 (UTC)
1808 ¦style=&quot;text-align:center;&quot;¦ 2015 ¦¦ '''[[Sylvester Stallone]]'''¦¦ ''[[Creed (film)¦Creed]]'' ¦¦ [[Rocky Balboa]]
1809 Sheehan moved to [[KNBC]] in 1984.  While at KNBC, he was the first local entertainment reporter to host and produce his own series of network specials, including &quot;Macho Men of the Movies&quot; (with [[Clint Eastwood]], [[Sylvester Stallone]] and [[Arnold Schwarzenegger]]) and &quot;Hollywood's Leading Ladies&quot; (with [[Julia Roberts]], [[Michelle Pfeiffer]], [[Sharon Stone]] and [[Barbra Streisand]]).{{citation needed¦date=July 2013}}  Sheehan returned to KCBS-TV in 1994 and remained with the station until 2004.&lt;ref&gt;[https://losangeles.cbslocal.com/2020/12/03/los-angeles-entertainment-journalist-david-sheehan-dies-82/ David Sheehan, Veteran LA Entertainment Journalist, Dies At 82] ''[[KCBS-TV]]/[[KCAL-TV]] (losangeles.cbslocal.com)''. 3 December 2020 (Retrieved 5 December 2020)&lt;/ref&gt;
1810 # 2008-01-27 02:46:47: User [[:en:User:71.119.250.52¦en:71.119.250.52]] ([[:en:User talk:71.119.250.52¦talk]] - [[:en:Special:Contributions/71.119.250.52¦contribs]]) to [[:en:Sylvester Stallone]] ([http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=187185492&amp;oldid=187185163 diff]). Links: www.sylvesterstallone.com www.filmreference.com/film/76/Sylvester-Stallone.html www.youtube.com/watch?v=FhGt77LGesQ www6.miami.edu/alumni/umaa/notable.htm video.google.com/videoplay?docid=-6493841613076981287 www..&lt;!--ID 1354977--&gt;
1811 # &lt;s&gt;2008-01-27 02:38:01: User [[:en:User:ClueBot¦en:ClueBot]] ([[:en:User talk:ClueBot¦talk]] - [[:en:Special:Contributions/ClueBot¦contribs]]) to [[:en:Sylvester Stallone]] ([http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=187184224&amp;oldid=187184214 diff]). Links: www.sylvesterstallone.com www.filmreference.com/film/76/Sylvester-Stallone.html www.youtube.com/watch?v=FhGt77LGesQ www6.miami.edu/alumni/umaa/notable.htm video.google.com/videoplay?docid=-6493841613076981287 www..&lt;!--ID 1354889--&gt;&lt;/s&gt;&lt;br /&gt;* User is on global [[:en:User:COIBot#Whitelist¦Whitelist]]: cluebot &lt;-&gt; * (Bots don't have a COI)
1812 # &lt;s&gt;2008-01-26 13:54:59: User [[:en:User:VoABot II¦en:VoABot II]] ([[:en:User talk:VoABot II¦talk]] - [[:en:Special:Contributions/VoABot II¦contribs]]) to [[:en:Sylvester Stallone]] ([http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=187070247&amp;oldid=187070240 diff]). Links: video.google.com/videoplay?docid=-6493841613076981287.&lt;!--ID 1346517--&gt;&lt;/s&gt;&lt;br /&gt;* User is on global [[:en:User:COIBot#Whitelist¦Whitelist]]: voabot ii &lt;-&gt; * (Bots don't have a COI)
1813 # 2007-12-20 11:43:19: User [[:en:User:Simon Bar Sinister¦en:Simon Bar Sinister]] ([[:en:User talk:Simon Bar Sinister¦talk]] - [[:en:Special:Contributions/Simon Bar Sinister¦contribs]]) to [[:en:Sylvester Stallone]] ([http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=179206045&amp;oldid=179183587 diff]). Links: video.google.com/videoplay?docid=-6493841613076981287.&lt;!--ID 984461--&gt;
1814 The documentary examines the steroid use of director Christopher Bell's two brothers, Mark &quot;Smelly&quot; Bell and [[Mike Bell (wrestler)¦Mike &quot;Mad Dog&quot; Bell]], who grew up idolizing [[Arnold Schwarzenegger]], [[Hulk Hogan]], and [[Sylvester Stallone]] (who all confessed to using steroids or growth hormones to achieve their physiques), and also features professional athletes, medical experts, fitness center members, and US Congressmen talking about the issue of [[anabolic steroid]]s.&lt;ref name=mc/&gt;
1815 [[Category:Images of Sylvester Stallone]]</text>
1816 *[[Honorary César]]:&lt;br/&gt;[[Michèle Morgan]]&lt;br/&gt;[[Sylvester Stallone]]
1817 * [[Rhinestone (film)¦''Rhinestone'' (film)]], a 1984 comedy starring Sylvester Stallone and Dolly Parton
1818 He appeared primarily in dubbing and is best known for dubbing over [[Sylvester Stallone]], [[Roy Scheider]] and [[Dean Martin]]. Note that he said in a 2018 interview that he has dubbed about 280 Hollywood stars so far.
1819 *'''[[Sylvester Stallone]]'''
1820 ¦ style=&quot;text-align: center;&quot; ¦&lt;ref&gt;{{Cite web¦url=https://www.msn.com/en-in/news/other/did-you-know-ulaganayagan-kamal-haasan-was-a-makeup-artist-for-sylvester-stallone-s-rombo-3/ar-BB1pcFBc¦title = Did you know Ulaganayagan Kamal Haasan was a makeup artist for Sylvester Stallone's Rombo 3?}}&lt;/ref&gt;
1821 The best-known person from the town is Alberto Aguilera Valadez, who is better known by his stage name, [[Juan Gabriel]]. Romualdo Bucío Bucío, whose stage name is [[Agustín Bernal]], was a prolific actor and director, whose influence on Mexican film has been compared to that of [[Arnold Schwarzenegger]] and [[Sylvester Stallone¦Sylvester Stalone]] to Hollywood. Actress [[Elpidia Carrillo]] was born in this town as well, coincidentally Elpidia is best known for her role as lead actress in [[Predator (film)¦Predator]] (1987) where she acted alongside Arnold Schwarzenegger.
1822     <redirect title="Sylvester Stallone" />
1823       <text bytes="116" sha1="kyj3xmikzpoqgjstqnt0t8roi9dpqjd" xml:space="preserve">#REDIRECT [[Sylvester Stallone]]
1824 ¦ ''[[Kambakkht Ishq]]'' ¦¦ [[Sabbir Khan]]  ¦¦ [[Akshay Kumar]], [[Kareena Kapoor]], [[Brandon Routh]], [[Vindu Dara Singh]], [[Aftab Shivdasani]], [[Amrita Arora]], [[Sylvester Stallone]], [[Javed Jaffrey]], [[Kirron Kher]], [[Boman Irani]] ¦¦ [[Action film¦Action]], [[Comedy film¦comedy]], [[Romance film¦romance]]
1825 # 2008-02-18 08:04:19: User [[:it:User:87.18.209.85¦it:87.18.209.85]] ([[:it:User talk:87.18.209.85¦talk]] - [[:it:Special:Contributions/87.18.209.85¦contribs]]) to [[:it:Sylvester Stallone]] ([http://it.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=14179253&amp;oldid=14148118&amp;rcid=14632841 diff]). Links: www.stallone.altervista.org/.&lt;!--ID 1662075--&gt;</text>
1826 Unrepresented by management, Davis sent his script out unsolicited to agents and studios over the course of the next three years. Davis secured representation through his efforts and received a number of studio offers,&lt;ref name = latimes&gt;{{cite news ¦ last = Gajewski ¦ first = Josh ¦ title = The Week Ahead; A fight film story to make Sly proud ¦ work =  Los Angeles Times ¦ date = 2005-05-02 }}&lt;/ref&gt; but, in a story reminiscent of [[Sylvester Stallone]]'s experience with ''[[Rocky]]'', he refused to sell the script unless he was signed to play the title role.&lt;ref name = chronicle&gt;{{cite news  ¦ last = Brown ¦ first = Joe ¦ title = Fighting Tommy Riley ¦ work = San Francisco Chronicle ¦ date = 2005-05-06 ¦ url = http://www.sfgate.com/cgi-bin/article.cgi?f=/c/a/2005/05/06/DDGIJCK97D1.DTL ¦ access-date = 2008-03-25}}&lt;/ref&gt; He also resisted demands that Marty be made heterosexual.&lt;ref name = gmax /&gt; While Davis and director O'Flaherty honed the script, Davis trained as a boxer to add to the film's authenticity.&lt;ref name = hinckle /&gt;&lt;ref&gt;{{cite news ¦ last = Seeley ¦ first = Darren ¦ title = Fighting Tommy Riley (2005) ¦ work = Choking on Popcorn ¦ date = 2005-10-14 ¦ url = http://www.chokingonpopcorn.com/popcorn/index.php?p=683 ¦ access-date = 2008-03-27 ¦ archive-date = 2009-08-29 ¦ archive-url = https://web.archive.org/web/20090829124145/http://www.chokingonpopcorn.com/popcorn/index.php?p=683 ¦ url-status = dead }}&lt;/ref&gt; The film was shot in Los Angeles on high-definition digital video&lt;ref name = hinckle /&gt; on a budget of $200,000 and was O'Flaherty's feature debut.&lt;ref name = latimes /&gt;
1827 [[Category:Video games based on works by Sylvester Stallone]]</text>
1828 # 2008-03-24 08:36:51: User [[:it:User:Antonfon¦it:Antonfon]] ([[:it:User talk:Antonfon¦talk]] - [[:it:Special:Contributions/Antonfon¦contribs]]; 21) to [[:it:Sylvester Stallone]] ([http://it.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=14966416&amp;oldid=14957704&amp;rcid=15450742 diff]) - Link: www.lastampa.it/_web/cmstp/tmplrubriche/giornalisti/grubrica.asp?ID_blog=69&amp;ID_articolo=1081&amp;ID_sezione=138&amp;sezione=Anteprime%20dagli%20Usa.&lt;!--ID 254799--&gt;
1829 ¦ [[Sylvester Stallone]]
1830 *Dwight &quot;The General&quot; Manfredi ([[Sylvester Stallone]]) - ''[[Tulsa King]]''
1831 In 1981, [[Sylvester Stallone]] commissioned Schomberg to create a bronze statue of [[Rocky Balboa]].&lt;ref name = &quot;profile&quot;/&gt; Three 2-ton, 10-foot copies were cast. One was installed atop the steps of the [[Philadelphia Museum of Art]] for the filming of ''[[Rocky III]]''.
1832 Martins was nicknamed ''Rambo'' and ''Stallone'', due to physical similarities with the [[Sylvester Stallone¦American actor]] and the [[John Rambo¦fictional character he portrayed]].&lt;ref name=FCB/&gt;
1833 After a long absence from the cinema, Monks made several appearances as an actor, beginning with [[Sylvester Stallone]]'s ''[[Paradise Alley]]'' (1978).
1834 Throughout his career, Lurie had a long-standing and well-known rivalry with his former business partner, [[Joe Weider]].  As a magazine publisher, bodybuilding  trainer and contest promoter, Lurie also had well publicized relationships with [[Arnold Schwarzenegger]] and [[Lou Ferrigno]].  These relationships both ended in litigation; the details of which were controversially detailed in Lurie's 2009 book ''Heart of Steel'', to which [[Regis Philbin]] contributed a foreword.  Throughout his colorful career, Lurie honored and befriended many Hollywood stars such as [[Mae West]], [[Steve Reeves]] and [[Clint Eastwood]].  In the course of his career in the fitness realm from the 1940s through the year 2000, Lurie owned and operated numerous Dan Lurie Gyms and Health Clubs in
```



```
New York and Miami Beach that attracted celebrities and bodybuilders alike. [[Sylvester Stallone]] famously trained at one of them.&lt;ref name=&quot;nytimes.com&quot;&gt;{{cite news |url=https://www.nytimes.com/2013/11/09/nyregion/dan-lurie-90-star-and-promoter-of-bodybuilding-dies.html?_r=0|title=Dan Lurie, 90, Star and Promoter of Bodybuilding, Dies|newspaper=New York Times|date=9 November 2013 |last1=Weber |first1=Bruce }}&lt;/ref&gt;&lt;ref&gt;{{cite web |publisher=Bodybudilding.com|url=http://www.bodybuilding.com/fun/drobson133.htm |title=Dan Lurie|date=15 December 2005 }}&lt;/ref&gt;
1835 *[[Sylvester Stallone]], actor&lt;ref&gt;[http://embeds.blogs.foxnews.com/2008/01/24/sylvester-stallone-endorses-mccain-coolest-endorsement-yet/ Sylvester Stallone Endorses McCain. Coolest Endorsement Yet?] {{Webarchive|url=https://web.archive.org/web/20080302161749/http://embeds.blogs.foxnews.com/2008/01/24/sylvester-stallone-endorses-mccain-coolest-endorsement-yet/ |date=2008-03-02 }}, ''FoxNews.com'', Retrieved January 24, 2008&lt;/ref&gt;
1836 [[Category:Images of Sylvester Stallone]]</text>
1837 *9 August - [[Alain Dorval]], voice actor of [[Sylvester Stallone]] (died [[2024 in France|2024]]).
1838 Flynn made a number of higher-profile films in the late 1980s, including the [[James Woods]] neo-noir ''[[Best Seller]]'', the [[Sylvester Stallone]] prison drama ''[[Lock Up (1989 film)|Lock Up]]'', and the [[Steven Seagal]] action film ''[[Out for Justice]]''.
1839 *[[Sylvester Stallone]] and [[Daryl Sabara]] Presented Favorite TV Show
1840 *''[[Rocky]]'' (1976)-Based on the screenplay by [[Sylvester Stallone]]&lt;ref name=&quot;Cascone&quot;/&gt;
1841 [[Jayam Ravi]] underwent physical training for the role of a forest officer in this film. He plays a one-man army in this film similar to that of [[Sylvester Stallone]] in ''[[Rambo (film series)|Rambo]]''.&lt;ref&gt;{{cite web|url=https://www.indiaglitz.com/peranmai-preview-telugu-movie-10060|title=Peranmai preview|website=Indiaglitz|access-date=25 July 2024}}&lt;/ref&gt;
1842 Celebrities impersonated include the populist and selfish former editor of the ''[[Daily Mirror]]'' newspaper [[Piers Morgan]], who gets a heavy object dropped on him each time; [[Victoria Beckham]], a big headed loudmouth with a speech disorder and her dumb husband [[David Beckham|David]], who is considerably shorter, going about their days working in America; [[Steven Spielberg]]; a transsexual version of [[Madonna]]; [[Morgan Freeman]]; large breasted model [[Jordan (Katie Price)|Jordan]] and child minded [[Peter Andre]]; old and senile action stars [[Harrison Ford]], [[Sylvester Stallone]] and [[Bruce Willis]], styled as &quot;The (Geriatric) Action Heroes&quot;, who fight their nemesis, money grabbing, snake haired old people hater [[Heather Mills]]; unintelligent [[Brad Pitt]] and his twisted wife [[Angelina Jolie]], who keeps adopting children and imprisoning them to a factory making hair extensions to rival the golden locks of [[Jennifer Aniston]], veteran newsreader [[Trevor McDonald]]; inappropriate uses for [[Russell Brand]] (e.g. as a toilet brush); [[Fabio Capello]] struggling to learn English leaving everyone confused at what he means when he speaks; far too similar footballers [[Steven Gerrard]] and [[Frank Lampard Jr.|Frank Lampard]], whose attempts to work together for England are portrayed as [[Laurel and Hardy]] style slapstick comedy; angry, loud mouthed [[Jeremy Clarkson]] making ill-informed judgements; [[Richard Hammond]] who was left drowning when global warming hit the 2050 episode of ''[[Top Gear (2002 TV series)|Top Gear]]''; sleepy, drunk and drug addled [[Pete Doherty]] and [[Amy Winehouse]]; complainers [[Bono]] and [[Bob Geldof]]; coin operated [[Jonathan Ross]]; bachelor [[George Clooney]]; [[Sebastian Coe]] presenting updates for the [[2012 Summer Olympics|2012 Olympics]]; [[Mohamed Al-Fayed]] and his conspiracy theories involving [[Fiat Uno#Fame and notoriety|Fiat Unos]]; alien [[Tom Cruise]] and his robot wife [[Katie Holmes|Katie]], and common chavs [[Helen Mirren]] and [[Judi Dench]], who bully [[Kate Winslet]] because they are Dames and she is not.
1843 [[Category:Cultural depictions of Sylvester Stallone]]
1844 | [[Sylvester Stallone]]
1845 In 1985, Beauvoir co-wrote and recorded several songs with [[Paul Stanley]] of the band [[Kiss (band)|Kiss]] for their thirteenth studio album, [[Asylum (Kiss album)|Asylum]].&lt;ref&gt;{{Citation|title=Asylum (Kiss album)|date=January 19, 2020|url=https://en.wikipedia.org/w/index.php?title=Asylum_(Kiss_album)&amp;oldid=936478859|work=Wikipedia|language=en|access-date=April 8, 2020}}&lt;/ref&gt; Beauvoir even played bass and provided backing vocals in [[Gene Simmons]]' absence on the songs &quot;Who Wants to Be Lonely&quot; and &quot;Uh! All Night.&quot; Following this, he launched a solo career with the album ''Drums Along the Mohawk'', released in the U.K. on [[Virgin Records]] in 1986. That same year, the track &quot;Feel the Heat&quot; was chosen by [[Sylvester Stallone]] for his film, ''[[Cobra (1986 film)|Cobra]]''.&lt;ref name=&quot;amg&quot; /&gt; The song was a hit, charting Top 10 across Europe and Australia and reaching No. 73 on the [[Billboard Hot 100|''Billboard'' Hot 100]].&lt;ref&gt;[{{AllMusic|class=artist|id=p55253/charts-awards/billboard-singles|pure_url=yes}} Billboard Singles], AllMusic&lt;/ref&gt; On the strength of the single, ''Drums Along the Mohawk'' was released in the U.S. and peaked at No. 93 on the [[Billboard 200|''Billboard'' 200]].&lt;ref&gt;[{{AllMusic|class=artist|id=p55253/charts-awards|pure_url=yes}} Billboard], AllMusic&lt;/ref&gt;
1846 Beauvoir produced and co-wrote the Ramones song &quot;My Brain Is Hanging Upside Down,&quot; originally titled &quot;Bonzo Goes To Bitburg.&quot; In addition to his film music work with Sylvester Stallone, his credit list includes the title tracks or theme songs for the movies ''[[Pet Sematary (1989 film)|Pet Sematary]],'' where he produced and arranged the Ramones' biggest radio hit, which reached No. 4 on Billboards Modern Rock Chart, ''[[Shocker (film)|Shocker]]'', ''[[Flawless (1999 film)|Flawless]]'', ''[[School of Rock]]'', ''[[Christmas with the Kranks]]'', where he produced and arranged the song Merry Christmas (I Don't Want To Fight Tonight), and performed, produced and wrote the end title song Merry Christmas To All of the World, ''[[The Guilty (2000 film)|The Guilty]]'', ''[[Rock 'n' Roll High School Forever]]'', ''Berlin Nights'', ''[[RV (film)|RV]]'',  ''[[Unthinkable]]'', and ''Why Him''?
1847 *&quot;Feel the Heat&quot; (Virgin Records, Columbia Records, Scotti Bros single 1986 lead track for Sylvester Stallone's ''Cobra'' film) Billboard [[Hot 100]] #73
1848 *Cobra (Sylvester Stallone) &quot;Feel the Heat&quot; (music producer, songwriter, performer and all instruments for lead song in film and trailer) 1986
1849 &quot;'''Whenever There Is Love'''&quot; is a song written by American singer and songwriter [[Bruce Roberts (singer)|Bruce Roberts]] and Edgar Bronfman Jr. (as Sam Roman) for the 1996 [[disaster film]] ''[[Daylight (1996 film)|Daylight]]'' starring [[Sylvester Stallone]].
1850 Cohen got his Hollywood start tech advising for numerous films including Steven Soderbergh's ''[[Haywire (2011 film)|Haywire]]'' and providing tactical training for actors including ''[[Keanu Reeves]]'' for the ''[[John Wick]]'' franchise, and quickly transitioned into acting, supporting in various film and TV projects including ''[[211]] with ''[[Nicolas Cage]]'', [[Rambo 5: Last Blood]]'' with [[Sylvester Stallone]] and a recurring role on the Netflix series Luis Miguel.  In 2022 he was cast in a supporting role opposite [[Antonio Banderas]] in the crime thriller [[The Enforcer (2022 film)|The Enforcer]]. &lt;ref&gt;{{cite news|url=https://deadline.com/2021/06/alexis-ren-antonio-banderas-kate-bosworth-2-chainz-movie-enforcer-starts-greece-1234783207/|title='The Enforcer': Mojean Aria, Zolee Griggs, Alexis Ren &amp; Aaron Cohen Join Antonio Banderas, Kate Bosworth &amp; 2 Chainz; Filming Begins In Greece |author=Andreas Wiseman|publisher=Deadline}}&lt;/ref&gt;
1851 The jury members of the contest are well-known film and television stars, athletes, entrepreneurs, and philanthropists. The jury and the honored guests of the Miss Ukraine Universe pageant: [[Naomi Campbell]], [[Dolph Lundgren]], [[Jean-Claude Van Damme]], [[Sylvester Stallone]], [[Jason Statham]], [[Joan Collins]], [[Thomas Anders]], [[Eric Trump]], [[Andriy Shevchenko|Andrii Shevchenko]], [[Snejana Onopka|Snizhana Onopko]], [[Alexandr Dolgopolov|Oleksandr Dolgopolov]], [[Caroline Wozniacki]], Paula Shugart, Miss Universe-2008 [[Dayana Mendoza]], Miss Universe-2005 [[Natalie Glebova]], [[Fadil Berisha]], [[Sherri Hill]] and others.
1852 *...that, inspired by [[Sylvester Stallone]]'s experience selling the script for ''[[Rocky]]'', actor/screenwriter [[J. P. Davis]] refused to sell his script for the film '''''[[Fighting Tommy Riley]]''''' unless he was guaranteed to play the lead?
1853 In attendance were celebrities such as [[Denzel Washington]], [[Fábio Coentrão]], [[Diego Milito]], [[Simon Bird]], [[Bruce Willis]], [[Sylvester Stallone]], [[Angelina Jolie]], [[Mario Lopez]], [[Brad Pitt]], [[Wesley Snipes]], [[Will Ferrell]], [[Gwen Stefani]], [[Jimmy Kimmel]], [[Tiger Woods]], [[Jude Law]], Jamie Hunt, [[Our Lady's Roman Catholic High School, Royton|David Lochery]], footballer [[David Beckham]], [[Kid Rock]], fellow [[Dancing with the Stars (American TV series)|Dancing with the Stars]] alums, [[Wayne Newton]], [[Mark Cuban]] and [[Dancing with the Stars (American season 5)|season 5]] winner [[Hélio Castroneves]] and [[Karina Smirnoff]], who is Mayweather's dance partner.&lt;ref&gt;{{cite web|url=http://www.laineygossip.com/Angelina_Jolie_Brad_Pitt_Gwen_Stefani_Gavin_Rossdale_at_Hatton_Mayweather_fight_in_Las_Vegas.aspx|title=The Pitts and King's parents|website=Laineygossip.com|access-date=7 December 2021}}&lt;/ref&gt; At the weigh in the crowd were whipped into a frenzy as [[Joe Calzaghe]] (who was there supporting Hatton) squared up to American legend [[Bernard Hopkins]] (who was there supporting Mayweather).&lt;ref&gt;{{Cite web|url=https://www.rickyhitmanhatton2.com/hopkins-calzaghe-confirmation-soon/|title=Hopkins-Calzaghe Confirmation soon |website=Rickyhitmanhatton2.com|date=14 September 2018|access-date=7 December 2021}}&lt;/ref&gt;
1854 | 2022 || ''[[Tulsa King]]'' || Vince Antonacci || Main role&lt;ref name=&quot;Mar22-cast&quot;&gt;{{cite web |last1=Andreeva |first1=Nellie |title=''Tulsa King'': Max Casella, Domenick Lombardozzi, Vincent Piazza &amp; Jay Will Join Sylvester Stallone In Paramount+ Series |url=https://deadline.com/2022/03/tulsa-king-max-casella-domenick-lombardozzi-vincent-piazza-jay-cast-sylvester-stallone-paramount-plus-1234985490/ |website=[[Deadline Hollywood]] |publisher=[[Penske Media Corporation]] |date=March 24, 2022}}&lt;/ref&gt;
1855 As ambassador, Rivkin made youth outreach one of his key priorities and connected the embassy to the next generation of leaders throughout France, including in disadvantaged communities in the ''[[banlieues]]'' outside larger cities.&lt;ref name = &quot;Le Parisien&quot;&gt;[http://www.leparisien.fr/espace-premium/val-d-oise-95/comment-les-etats-unis-investissent-les-cites-26-01-2013-2513079.php &quot;Comment les Etats-Unis investissent les cités&quot;], ''Le Parisien'', January 26, 2013.&lt;/ref&gt; Rivkin organized a series of seminars for French youth, inviting them to meet with prominent American government officials, actors and musicians. Beginning with actor [[Samuel L. Jackson]]'s April 2010 visit with students in [[Bondy]], an economically depressed Parisian suburb, Rivkin set up seminars and hosted events with [[Stephen Colbert]], [[Sylvester Stallone]], [[Woody Allen]], [[Jodie Foster]], [[will.i.am]] of the [[Black Eyed Peas]], [[Robert Zemeckis]], [[Allen Stone]], [[Tony Bennett]], [[Herbie Hancock]], [[Maya Angelou]], [[Toni Morrison]] and many others.&lt;ref name=&quot;OIG Report&quot; /&gt;&lt;ref name=&quot;NY Times Outreach&quot;&gt;{{cite news|last=Sayare|first=Scott|title=Feeling Slighted by France, and Respected by the U.S.|url=https://www.nytimes.com/2010/09/23/world/europe/23france.html|newspaper=The New York Times|date=Sep 22, 2010 |url-access=subscription |url-status=live |archive-url=https://web.archive.org/web/20221207215227/https://www.nytimes.com/2010/09/23/world/europe/23france.html |archive-date= Dec 7, 2022 }}&lt;/ref&gt; &quot;Much of the embassy's outreach is meant to dispel 'mistruths' about the United States,&quot; Rivkin said in an interview, adding, &quot;It's easier to hate something you don't understand.&quot;&lt;ref name=&quot;NY Times Outreach&quot; /&gt; In January 2012, Rivkin broadened his outreach efforts by creating the Washburne Award for Innovation in Diversity, recognizing one French and one American company for their best practices in fostering diversity in hiring practices.&lt;ref name = &quot;Le Monde&quot;&gt;[http://www.lemonde.fr/economie/article/2013/05/30/un-plan-coordonne-en-faveur-de-l-innovation-un-projet-politique-global-sera-propose-a-la-mi-juillet_3421059_3234.html &quot;Un plan coordonné en faveur de l'innovation | Nos amis américains | Prix Washburne&quot;], ''Le Monde'', May 30, 2013.&lt;/ref&gt;
1856 It also stars many Hollywood artists including [[Sylvester Stallone]], [[Martin Short]], [[Alicia Silverstone]], [[Charlie Sheen]], [[Hugh Grant]], [[Ed Harris]], [[Gérard Depardieu]], [[Jack Nicholson]], [[Brooke Shields]], [[Matthew McConaughey]], [[Ashley Judd]], [[Sharon Stone]], [[John Travolta]], [[Kelly Preston]], [[Emma Thompson]], [[Arnold Schwarzenegger]], [[Tom Cruise]], [[Nicolas Cage]], [[Clint Eastwood]], [[Jim Carrey]], [[Robert De Niro]], [[Whoopi Goldberg]], [[Robin Williams]], [[Richard Gere]], [[Antonio Banderas]], [[Pauly Shore]], [[Sean Connery]], [[Julianne Moore]], [[Madonna (entertainer)|Madonna]], and [[Anthony Hopkins]]. It was produced by Rick Caine.
1857 Johnny Strong has had various roles in television, and landed a role in the [[Steven Seagal]] film ''[[The Glimmer Man]]''. He then worked on [[Sylvester Stallone]]'s ''[[Get Carter (2000 film)|Get Carter]]'', ''[[The Fast and the Furious (2001 film)|The Fast and The Furious]]'', and ''[[Black Hawk Down (film)|Black Hawk Down]]''. Strong stars in ''[[Sinners &amp; Saints (film)|Sinners &amp; Saints]]'', a New Orleans-based police thriller featuring [[Sean Patrick Flanery]] and [[Tom Berenger]] released in 2012.
1858 '''''[[Grudge Match]]''''' is a 2013 film starring Sylvester Stallone and Robert De Niro.
```



1859 <redirect title="Sylvester Stallone" />

1860 <comment>[[WP:AES|←]]Redirected page to [[Sylvester Stallone]]</comment>

1861 <text bytes="32" sha1="p7gz58bc0f4qqxa7url52a8fje2gfcq" xml:space="preserve">#REDIRECT [[Sylvester Stallone]]</text>

1862 | [[Sylvester Stallone]], [[Richard Crenna]], [[Charles Napier (actor)|Charles Napier]], [[Julia Nickson]] and [[Steven Berkoff]]

1863 | [[Sylvester Stallone]], [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], [[Brigitte Nielsen]] and [[Dolph Lundgren]]

1864 | Sylvester Stallone

1865 On April 25, 2019, ''A Sense of Purpose: Fighting For Our Lives'' premiered at the Start Cinema &amp; Cafe in Brooklyn, NY.&lt;ref&gt;{{cite web |url=https://www.rfdtv.com/story/40337907/new-film-brings-awareness-to-ptsd-veterans-and-military-sexual-trauma-in-april-sexual-assault-awareness-month |title=New Film Brings Awareness to PTSD, Veterans and Military Sexual Trauma in April, Sexual Assault Awareness Month |work=Rfdtv.com |access-date=May 24, 2019}}&lt;/ref&gt; Colonel Annette Tucker Osborne, President of The National Association of Black Military Women, Brooklyn Chapter awarded Quinlan the [[Challenge coin]] for his role in the film.&lt;ref&gt;{{cite web |url=https://www.blogtalkradio.com/conversationslive/2019/10/29/model-and-actor-john-j-quinlan-returns-to-conversationslive |title=Model and Actor John J. Quinlan Returns to Conversations Live |work=Blogtalkradio.com |access-date=October 31, 2019}}&lt;/ref&gt; During an interview with Conversations Radio Network President Cyrus Webb at the event, Quinlan credited [[Sylvester Stallone]] as his biggest inspiration as an actor.&lt;ref&gt;{{cite web |url=https://www.youtube.com/watch?v=hS9AciXZv50 | archive-url=https://ghostarchive.org/varchive/youtube/20211221/hS9AciXZv50 |archive-date=2021-12-21 |url-status=live|title=Actor John Quinlan Talks A Sense of Purpose on Cyrus Webb Presents In New York |work=Youtube.com |access-date=June 8, 2019}}{{cbignore}}&lt;/ref&gt;

1866 *[[Sylvester Stallone]]&lt;ref&gt;{{cite web|url=https://allmovie.com/artist/sylvester-stallone-112464|work=Allmovie|access-date=March 4, 2010|title=Sylvester Stallone: Overview - Allmovie| author=Brennan, Sandra}}&lt;/ref&gt;

1867 In 1945, Amendola made his voice-over debut dubbing over Vito Annicchiarico's role in the film ''[[Rome, Open City]]''. By 1968, Amendola began devoting most of his time to voice-over acting and eventually became well known as an Italian voice dubbing pioneer.&lt;ref&gt;[https://www.antoniogenna.net/doppiaggio/speciali/cronologia.htm Cronologia fondamentale dell'epoca d'oro del doppiaggio italiano Dagli albori agli anni 1970] (in Italian)&lt;/ref&gt; He was best known as the Italian voice of famous actors including [[Al Pacino]], [[Sylvester Stallone]], [[Dustin Hoffman]], [[Robert De Niro]] and [[Tomas Milian]] in a majority of their movies.&lt;ref&gt;{{cite web |url=https://popcorntv.it/guide/attori-doppiati-da-ferruccio-amendola/32593 |title=Attori doppiati da Ferruccio Amendola |publisher=news.popcorn.it |date= |access-date=16 March 2020}}&lt;/ref&gt; He also dubbed [[Peter Falk]] as well as [[Bill Cosby]] in the television series ''[[The Cosby Show]]'' as well as ''[[Cosby (TV series)|Cosby]]''. In addition, he has done the voice over in commercials and TV dramas with great success.&lt;ref&gt;{{cite web |url=https://www.ecodelcinema.com/ferruccio-amendola-biografia-filmografia.htm |website=ecodelcinema.com |title=Ferruccio Amendola - Attore - Biografia e Filmographia - Ecodelcinema |date=29 June 2014 |access-date=17 March 2020}}&lt;/ref&gt;

1868 |[[Sylvester Stallone]]

1869 |Sylvester Stallone

1870 |Sylvester Stallone

1871 |rowspan=2|Sylvester Stallone

1872 |rowspan=2|Sylvester Stallone

1873 |Sylvester Stallone

1874 |rowspan=2|Sylvester Stallone

1875 |Sylvester Stallone

1876 |Sylvester Stallone

1877 |Sylvester Stallone

1878 |rowspan=2|Sylvester Stallone

1879 |Sylvester Stallone

1880 |Sylvester Stallone

1881 |Sylvester Stallone

1882 |rowspan=2|Sylvester Stallone

1883 |Sylvester Stallone

1884 |rowspan=2|Sylvester Stallone

1885 |Sylvester Stallone

1886 |Sylvester Stallone

1887 |Sylvester Stallone

1888 |Sylvester Stallone

1889 Then and Now 2006: [[Sylvester Stallone]] (Now: 2006 - ''[[Rocky Balboa (film)|Rocky Balboa]]'' / Then: 1980s - ''[[Rocky III]]'' and ''[[Rocky IV]]'') and [[William Shatner]] (Now: 2006 - [[Priceline.com]] / Then: 1980s - ''[[T.J. Hooker]]'')

1890 | [[Sylvester Stallone]]

1891 | [[Sylvester Stallone]], [[Talia Shire]], [[Burt Young]], [[Carl Weathers]], [[Burgess Meredith]] and [[Mr. T]]

1892 | Sylvester Stallone

1893 It features [[Akshay Kumar]] and [[Kareena Kapoor]] in pivotal roles along with [[Aftab Shivdasani]], [[Amrita Arora]], [[Jaaved Jaaferi|Jaaved Jaafrey]] and [[Kirron Kher]] in supporting roles. [[Sylvester Stallone]], [[Denise Richards]], [[Brandon Routh]] and [[Holly Valance]] appear in cameos, playing themselves.&lt;ref name=&quot;Filming&quot;&gt;{{cite news | author=Bollywood Hungama News Network | title=Sylvester Stallone shoots for Kambakht Ishq | url=https://www.bollywoodhungama.com/news/slideshows/slideshow-all-that-will-happen-on-kbc-this-holi/2008/ | work=[[Bollywood Hungama|IndiaFM]] | access-date=2008-07-24 | url-status=live | archive-url=https://web.archive.org/web/20090204161328/http://www.bollywoodhungama.com/slideshow/2008/07/24/263/ | archive-date=4 February 2009}}&lt;/ref&gt;

1894 * [[Sylvester Stallone]] as himself

1895 [[Category:Images of Sylvester Stallone]]</text>

1896 [[Category:Images of Sylvester Stallone]]</text>

1897 * Krugarr makes a non-speaking appearance in the [[Marvel Cinematic Universe]] (MCU) film ''[[Guardians of the Galaxy Vol. 2]]''.&lt;ref&gt;{{cite web|title=Guardians of the Galaxy Vol. 2's Doctor Strange Connection|url=http://comicbook.com/marvel/2017/05/06/guardians-galaxy-2-doctor-strange-connection-krugarr/#2|website=Marvel}}&lt;/ref&gt;&lt;ref name=&quot;OldTeam&quot;&gt;{{cite web| last=Daniell|first=Mark|date=April 25, 2017|title=Sylvester Stallone's 'Guardians of the Galaxy Vol. 2' character has a big future in the MCU|url=http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu|archive-url=https://web.archive.org/web/20170428051803/http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu|archive-date=April 28, 2017|access-date=April 28, 2017|work=The Whig|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|author=|date=8 May 2017|title=Marvel's Guardians of the Galaxy: Get To Know the Original Cosmic Team|url=https://www.cbr.com/marvel-original-guardians-galaxy-team-explained/|access-date=28 December 2018| website=CBR}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last=Polo|first=Susana|date=4 May 2017|title=Guardians of the Galaxy Vol. 2's five different credits scenes, explained|url=https://www.polygon.com/2017/5/4/15468856/guardians-of-the-galaxy-vol-2-post-credits-scenes-explained|access-date=28 December 2018|website=Polygon}}&lt;/ref&gt;&lt;ref&gt;{{cite web| last=Acuna|first=Kirsten|date=|title=Here's who that mystery group is at the end of 'Guardians of the Galaxy Vol. 2'|url=https://www.thisisinsider.com/guardians-of-the-galaxy-sylvester-stallone-role-explained-2017-5|access-date=28 December 2018|website=INSIDER}}&lt;/ref&gt; This version is a member of the [[Ravagers (Marvel Cinematic Universe)|Ravagers]] and an associate of [[Yondu Udonta (Marvel Cinematic Universe)|Yondu Udonta]]. Following Udonta's death, Krugarr meets with [[Stakar Ogord (Marvel Cinematic Universe)|Stakar Ogord]], [[Martinex (Marvel Cinematic Universe)|Martinex T'Naga]], [[Charlie-27]], [[Aleta Ogord]], and [[Mainframe (comics)|Mainframe]] to reunite their original group in Udonta's memory.&lt;ref&gt;{{cite web|date=20 April 2017|title=Adam Warlock &amp; Original Guardians Join Marvel Cinematic Universe|url=http://movieweb.com/adam-warlock-guardians-galaxy-marvel-cinematic-universe-mcu/|publisher=}}&lt;/ref&gt;

1898 | [[Sylvester Stallone]], [[Sung Kang]]

1899 [[Category:Images of Sylvester Stallone]]</text>

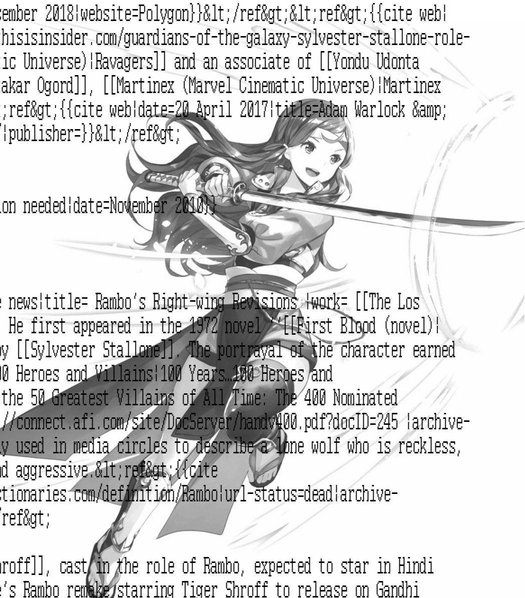
1900 The plot and title of the film are derived from the 1986 film [[Cobra (1986 film)|''Cobra'']] starring [[Sylvester Stallone]].{{Citation needed|date=November 2010}}

1901 | caption = [[Sylvester Stallone]] as John Rambo in &lt;br&gt; ''[[Rambo III]]'' (1988)

1902 | portrayer = [[Sylvester Stallone]]

1903 * Sylvester Stallone &lt;br /&gt; (''[[Mortal Kombat 11]]'')

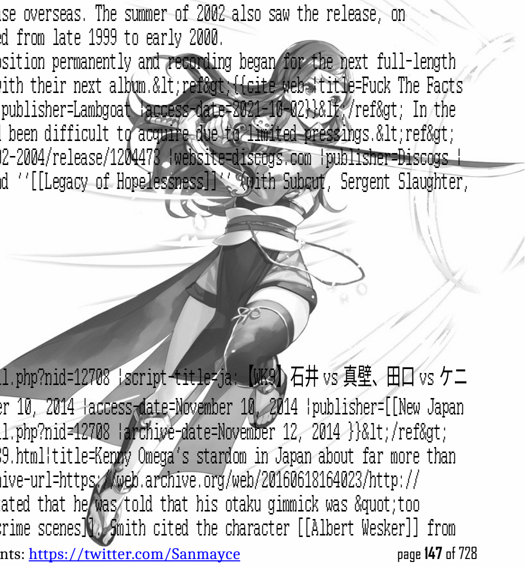
1904 '''John James Rambo''' (born July 6, 1947) is a fictional character in the [[Rambo (franchise)|''Rambo'' franchise]].&lt;ref&gt;{{cite news|title= Rambo's Right-wing Revisions |work= [[The Los Angeles Times]]|url= https://www.latimes.com/archives/la-xpm-1985-07-14-ca-6307-story.html|access-date=December 30, 2010}}&lt;/ref&gt; He first appeared in the 1972 novel ''[[First Blood (novel)| First Blood]]'' by [[David Morrell]], but later became more famous as the [[protagonist]] of the film series, in which he was played by [[Sylvester Stallone]]. The portrayal of the character earned Stallone widespread acclaim and recognition. The character was nominated for [[American Film Institute]]'s list ''[[AFI's 100 Years…100 Heroes and Villains|100 Years…100 Heroes and Villains]]''.&lt;ref&gt;{{cite web|url=http://connect.afi.com/site/DocServer/handv400.pdf?docID=245 |title=The 50 Greatest Heroes and the 50 Greatest Villains of All Time: The 400 Nominated Characters |publisher=afi.com |access-date=May 21, 2010 |url-status=dead |archive-url=https://web.archive.org/web/20110807135603/http://connect.afi.com/site/DocServer/handv400.pdf?docID=245 |archive-date=August 7, 2011 }}&lt;/ref&gt; Following the success of the first movie, the term &quot;[[wikt:Rambo|Rambo]]&quot; was occasionally used in media circles to describe a lone wolf who is reckless, disregards orders, uses violence to solve all problems, enters dangerous situations alone, and is exceptionally tough, callous, raw and aggressive.&lt;ref&gt;{{cite web|url=https://en.oxforddictionaries.com/definition/Rambo|archive-url=https://web.archive.org/web/20171025184435/https://en.oxforddictionaries.com/definition/Rambo|url-status=dead|archive-date=October 25, 2017|title=Rambo - Definition of Rambo in English by Oxford Dictionaries|website=Oxford Dictionaries - English}}&lt;/ref&gt;



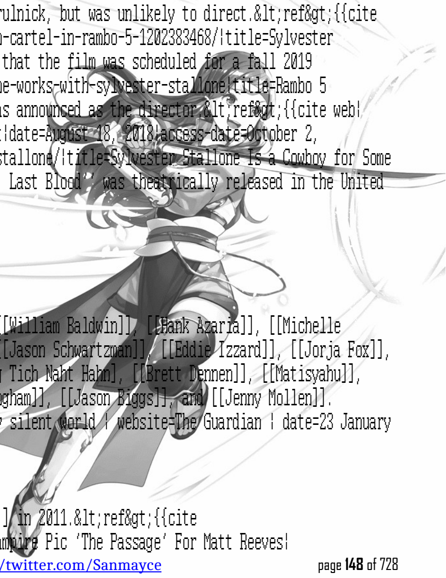
1905 In all five films, Rambo is portrayed by [[Sylvester Stallone]]. In the animated TV series, the character is voiced by [[Neil Ross]].

1906 A [[Bollywood]] remake of ''First Blood'' {{as of|2019||alt=was scheduled as of 2019}} to be released in October 2020, with [[Tiger Shroff]], cast in the role of Rambo, expected to star in Hindi remakes of all five films in the ''Rambo'' franchise.&lt;ref&gt;{{cite news |last1=Dixit |first1=Ayush Mohan |title=Sylvester Stallone's Rambo remake starring Tiger Shroff to release on Gandhi

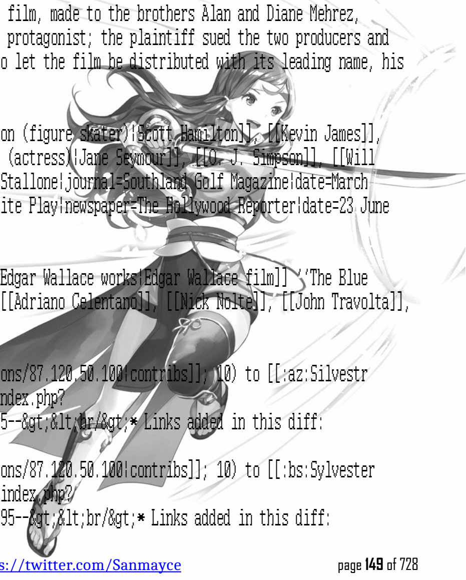
```
     Jayanti 2020 {{!}} Bollywood News |url=https://www.timesnownews.com/entertainment/news/bollywood-news/article/sylvester-stallone-s-rambo-remake-starring-tiger-shroff-to-release-on-gandhi-jayanti/
     420401 |access-date=May 31, 2019 |work=[[Times Now]] |date=May 17, 2019}}&lt;/ref&gt;
1907 During the [[2019 Cannes Film Festival]], Stallone said he would continue portraying Rambo if the fifth film succeeded.&lt;ref&gt;{{cite web|url=https://www.joblo.com/horror-movies/news/stallone-
     would-return-as-rambo-if-last-blood-is-a-hit|title= Stallone happy to return as Rambo if Last Blood is a hit|first=Mike|last=Sprague|date=June 12, 2019|publisher=[[JoBlo.com]]|access-date=June 13,
     2019}}&lt;/ref&gt; Grunberg, however, said that ''Rambo: Last Blood'' &quot;closes the circle&quot;, hoping it would conclude the film series.&lt;ref name=&quot;Total Film&quot;&gt;{{cite web|
     url=https://www.gamesradar.com/rambo-5-last-blood-director-adrian-grunberg-interview/|title= Exclusive: Rambo: Last Blood director discusses bringing back Sylvester Stallone's action hero: &quot;This
     movie closes the circle&quot;|author=Jack Shepherd, James Mottram|work=[[GamesRadar]]|date=July 22, 2019|access-date=July 25, 2019}}&lt;/ref&gt; In September 2019, Stallone confirmed that he has
     plans for a prequel to the series; although he would not reprise the title role, he would like to explore who Rambo was before the war: {{blockquote|I always thought of Rambo when he was 16 or 17-I
     hope they can do the prequel-he was the best person you could find. He was the captain of the team; he was the most popular kid in school; super athlete. He was like [[Jim Thorpe]], and the war is
     what changed him. If you saw him before, he was like the perfect guy.&lt;ref&gt;{{cite web|last=Zinski|first=Dan |title=Sylvester Stallone Wants a Rambo Prequel|url=https://screenrant.com/rambo-
     prequel-movie-idea-sylvester-stallone/ |website=[[Screen Rant]] |date=September 16, 2019 |access-date=17 September 2019}}&lt;/ref&gt;|sign=|source=}}
1908 Stallone has expressed interest in having Rambo take refuge in an [[Indian reservation]] for the sixth ''Rambo'' film.&lt;ref&gt;{{cite web|title=Sylvester Stallone interview for Rambo: Last Blood!|
     url=https://www.youtube.com/watch?v=2th50Z9nKcI |archive-url=https://ghostarchive.org/varchive/youtube/20211221/2th50Z9nKcI |archive-date=2021-12-21 |url-status=live|work=[[JoBlo.com]]|
     date=September 17, 2019 |publisher=JoBlo.com official YouTube channel|access-date=September 18, 2019|time=3:52}}{{cbignore}}&lt;/ref&gt; In June 2020, Stallone briefly elaborated on the idea,
     stating, &quot;If I ever did another one, I think he would go back to the Indian reservation that he grew up on because he has Indian family. As of 2022, there will be no Rambo 6. &quot;&lt;ref&gt;
     {{Cite web|url=https://www.youtube.com/watch?v=otyZ9qS3-cY |archive-url=https://ghostarchive.org/varchive/youtube/20211221/otyZ9qS3-cY |archive-date=2021-12-21 |url-status=live|title=ランボー ラスト・
     ブラッド シルベスター・スタローン主演 インタビュー最後の勇者編 Rambo: Last Blood Sylvester Stallone interview movie|date=June 23, 2020|work=[[Nippon TV]] Sukkiri|publisher=Shigz Channel, YouTube|
     access-date=June 25, 2020|time=8:30}}{{cbignore}}&lt;/ref&gt;
1909 * [[Rambo (2008 film)|''Rambo'']] (2008) - the sequel to the 1988 film, directed by [[Sylvester Stallone]]
1910 * ''First Kill'', in November 2022 an Indiegogo campaign launched for a new graphic novel depicting Rambo's first tour of duty in Vietnam written by Sylvester Stallone and Chuck Dixon.
1911 * ''[[Mortal Kombat 11]]'', a [[fighting game]] in which Rambo is a playable DLC character via [[downloadable content]], with Sylvester Stallone reprising his role.&lt;ref&gt;{{cite web |title=Mortal
     Kombat 11 Ultimate brings Rain, Mileena, and Rambo |url=https://blog.playstation.com/2020/10/08/mortal-kombat-11-ultimate-brings-rain-mileena-and-rambo/ |publisher=PlayStation Blog |access-
     date=October 8, 2020 |date=October 8, 2020}}&lt;/ref&gt; In his arcade ending, Rambo defeated Kronika and planned to use her hourglass to right every wrong in history. As picking who lives and who
     dies would &quot;kill his soul&quot;, Rambo walked away for good this time while stating &quot;After all I've been through, I've earned a little peace&quot;.
1912 John Rambo is considered a [[cultural icon]].&lt;ref&gt;{{cite web|last=Saathoof|first=Evan|title=RAMBO And The All-Too-Occasional Brilliance Of Sylvester
     Stallone|url=https://birthmoviesdeath.com/2015/04/14/rambo-and-the-all-too-occasional-brilliance-of-sylvester-stallone|date=April 14, 2015|website=Birthmoviesdeath.com}}&lt;/ref&gt;&lt;ref&gt;{{cite
     book|last=Enloe|first=Cynthia H.|title=The Morning After: Sexual Politics at the End of the Cold War|location=Berkeley and Los Angeles, California|publisher=University of California Press|year=1993|
     url=https://books.google.com/books?id=tHTECm_g6xoC&amp;q=The%20Morning%20After%3A%20Sexual%20Politics%20at%20the%20End%20of%20the%20Cold%20War&amp;pg=PP1|isbn=0520914104}}&lt;/ref&gt;&lt;ref&gt;
     {{cite book|last=Holt|first=Douglas B.|title=How Brands Become Icons: The Principles of Cultural Branding|publisher=Harvard Business Press|location=Brighton,
     Massachusetts|year=2004|url=https://books.google.com/books?id=PqfSW3zpvFoC&amp;q=How%20Brands%20Become%20Icons%3A%20The%20Principles%20of%20Cultural%20Branding&amp;pg=PR1|isbn=1422163326}}&lt;/
     ref&gt;&lt;ref&gt;{{cite book|editor-last1=Matelski|editor-first1=Marilyn J.|editor-last2=Lynch Street|editor-first2=Nancy|title=War and Film in America: Historical and Critical Essays|
     publisher=McFarland &amp; Company|location=Jefferson, North Carolina|year=2003|url=https://books.google.com/books?id=EUGNAgAAQBAJ&amp;q=War%20and%20Film%20in%20America%3A%20Historical%20and
     %20Critical%20Essays&amp;pg=PR3|isbn=0786416734}}&lt;/ref&gt;&lt;ref&gt;{{cite news|last1=Mendelson|first1=Scott|title='Rambo: First Blood Part II' Was The 'American Sniper' Of Its
     Day|url=https://www.forbes.com/sites/scottmendelson/2015/05/22/rambo-first-blood-part-ii-was-the-american-sniper-of-its-day/#2111706a50b9|date=May 22, 2015|magazine=Forbes|issn=0015-6914|
     oclc=892320816}}&lt;/ref&gt; The character influenced many action heroes and films in the 1980s and 1990s. The John Rambo character became a prominent part of pop culture, with the term
     &quot;Rambo&quot; becoming a descriptor for individuals or situations. For example in the legal profession a &quot;Rambo lawyer&quot; is one who habitually engages in &quot;all manners of adversarial
     excess, including personal attacks on other lawyers, hostility, boorish and insulting behavior, rudeness and obstructionist conduct&quot;&lt;ref&gt;Ronald Hicks, Jr.
     &quot;[http://www.insidecounsel.com/2012/07/26/litigation-8-tips-for-dealing-with-a-rambo-lawyer 8 tips for dealing with a Rambo lawyer: How to handle personal attacks or underhanded litigation
     tactics from opposing counsel] {{Webarchive|url=https://web.archive.org/web/20150927212949/http://www.insidecounsel.com/2012/07/26/litigation-8-tips-for-dealing-with-a-rambo-lawyer |date=September
     27, 2015 }}.&quot; July 26, 2012, InsideCounsel.com, accessed December 10, 2015&lt;/ref&gt; or embraces &quot;a 'take no prisoners' attitude&quot;.&lt;ref&gt;Thomas M. Reavley (1990)
     &quot;[http://digitalcommons.pepperdine.edu/cgi/viewcontent.cgi?article=1730&amp;context=plr Rambo Litigators: Pitting Aggressive Tactics Against Legal Ethics]&quot;. Pepperdine Law Review, Vol 17.
     No 3&lt;/ref&gt; American Football player [[Mark Bavaro]], who played professionally in the [[National Football League|NFL]] from 1985 to 1995,&lt;ref name=&quot;jeans2&quot;&gt;Litsky, Frank.
     [https://select.nytimes.com/search/restricted/article?res=F50811FE3E5F0C728CDDA00894DD484D81 Mowatt Sidelined for Entire Season], ''The New York Times'', September 1, 1985, accessed May 14,
     2007.&lt;/ref&gt;&lt;ref name=&quot;dbf&quot;&gt;[http://www.databasefootball.com/players/playerpage.htm?ilkid=BAVARMAR01 Mark Bavaro]
     {{webarchive|url=https://web.archive.org/web/20060328102824/http://www.databasefootball.com/players/playerpage.htm?ilkid=BAVARMAR01|date=March 28, 2006}}, databasefootball.com, accessed May 14,
     2007.&lt;/ref&gt; was nicknamed Rambo during his career.&lt;ref&gt;{{cite web|last=Serby|first=Steve|date=August 24, 2017|title=Giants 'Rambo' legend is drooling over Evan
     Engram|url=https://nypost.com/2017/08/24/giants-rambo-legend-is-drooling-over-evan-engram/}}&lt;/ref&gt; [[Sitiveni Rabuka]], the current [[prime minister of Fiji]] who led two coups against the
     Fijian government in 1987, is also nicknamed &quot;Rambo.&quot;
1913 {{Sylvester Stallone}}
1914 [[Category:Sylvester Stallone]]
1915 ** [[Sylvester Stallone]] - ''[[Rocky]]'' as [[Rocky Balboa]]
1916 ** ''[[Rocky]]'' - [[Sylvester Stallone]]
1917 Director Doug Liman stated that he had been a fan of ''[[The Bourne Identity (novel)|The Bourne Identity]]'' by Robert Ludlum since he read it in high school. Near the end of production of Liman's
     previous film ''[[Swingers (1996 film)|Swingers]]'', Liman decided to develop a film adaptation of the novel. After more than two years of securing rights to the book from [[Warner Brothers]] and a
     further year of screenplay development with screenwriter Tony Gilroy, the film went through two years of production.&lt;ref name=&quot;DVD.com&quot;&gt;''The Bourne Identity'' DVD Commentary
     Featuring Doug Liman (2003).&lt;/ref&gt; Liman approached a wide range of actors for the role of Bourne, including [[Russell Crowe]] and [[Sylvester Stallone]], before he eventually cast [[Matt
     Damon]]. Liman found that Damon understood and appreciated that, though ''The Bourne Identity'' would have its share of action, the focus was primarily on character and plot.&lt;ref
     name=&quot;BBC&quot;&gt;{{cite web |last=Hanrahan |first=Denise |work=BBC.co.uk |title=Interview with Doug Liman
     |url=https://www.bbc.co.uk/films/2002/09/05/doug_liman_the_bourne_identity_interview.shtml |access-date=March 14, 2007}}&lt;/ref&gt; Production was difficult, with screenplay rewrites occurring
     throughout the entire filming and Liman constantly arguing with Universal's executives.&lt;ref name=&quot;Wall&quot;&gt;{{cite web |last=King |first=Tom |work=Wall Street Journal |title=Bourne to be
     Wild |url=http://www.murphsplace.com/owen/articles/journal2.html |access-date=March 12, 2007 |url-status=dead |archive-url=https://web.archive.org/web/20121015074908/http://www.murphsplace.com/owen/
     articles/journal2.html |archive-date=October 15, 2012 |df=mdy-all }}&lt;/ref&gt; ''[[The Bourne Identity (2002 film)|The Bourne Identity]]'' was released in June 2002.
1918 Ferreux would go on to appear in a number of roles as an adult in his native France in film and television. In 1981, he appeared in the [[John Huston]]-directed English language drama ''[[Escape to
     Victory]]'' with actors [[Sylvester Stallone]], [[Michael Caine]], [[Max von Sydow]] and Brazilian footballer [[Pelé]].
1919 The summer of 2002 saw the band expand their touring area reaching as far west as [[Winnipeg]], and including a December tour of the [[Maritimes]]. The touring, along with the release of a split with
     Sylvester Staline,&lt;ref&gt;{{cite web |title=Sylvester Staline / Fuck The Facts - We Fuck The Facts / Fuck The Facts |url=https://www.discogs.com/Sylvester-Staline-Fuck-The-Facts-We-Fuck-The-Facts-
     Fuck-The-Facts/release/1317340 |website=discogs.com |publisher=Discogs |access-date=2021-10-02}}&lt;/ref&gt; helped expand their fanbase overseas. The summer of 2002 also saw the release, on
     Mandarangan Recordings, of ''[[Escunta]]'', the long-awaited noise follow-up to ''[[Vagina Dancer]]'', which had actually been recorded from late 1999 to early 2000.
1920 At the start of 2006, Tim Olsen was asked to leave the band. Mathieu Vilandré, who had previously filled in on drums, took over the position permanently and recording began for the next full-length
     album. Early in the recording process, the band was contacted by [[Relapse Records]] and secured a three-album deal that would start with their next album.&lt;ref&gt;{{cite web |title=Fuck The Facts
     sign w/ Relapse Records, April 2006 |url=https://lambgoat.com/news/6588/Fuck-The-Facts-sign-w--Relapse-Records |website=lambgoat.com |publisher=Lambgoat |access-date=2021-10-02}}&lt;/ref&gt; In the
     interim, the ''[[Collection of Splits 2002-2004]]'' was released by Great White North Records, and exposed fans to some music that had been difficult to acquire due to limited pressings.&lt;ref&gt;
     {{cite web |title=Fuck The Facts - Collection Of Splits 2002-2004 |url=https://www.discogs.com/Fuck-The-Facts-Collection-Of-Splits-2002-2004/release/1204473 |website=discogs.com |publisher=Discogs |
     access-date=2021-10-02}}&lt;/ref&gt; It included all of the material from the splits released between ''[[Backstabber Etiquette]]'' and ''[[Legacy of Hopelessness]]'' (with Subcut, Sergent Slaughter,
     Feeble Minded and Sylvester Staline) as well as enhanced MP3 tracks from some earlier splits.
1921 [[Category:Images of Sylvester Stallone]]</text>
1922 |[[Sylvester Stallone]]
1923 | extra18          = [[Fuck the Facts splits#Split with Sylvester Staline|Split with Sylvester Staline]]
1924 | extra19          = Split with Sylvester Staline
1925 | extra20          = Split with Sylvester Staline
1926 | extra21          = Split with Sylvester Staline
1927 During his run in Bullet Club, Smith did his interviews entirely in English,&lt;ref&gt;{{cite web|url=http://www.njpw.co.jp/news/detail.php?nid=12708 |script-title=ja:【WK9】石井 vs 真壁、田口 vs ケニ
     ーが決定!「暴走キングコングを眠りから起こす!」（石井）、「グッドガイの私はもういない!」（ケニー）【1.4 東京ドーム会見】|date=November 10, 2014 |access-date=November 10, 2014 |publisher=[[New Japan
     Pro-Wrestling]] |language=ja |url-status=dead |archive-url=https://web.archive.org/web/20141112224440/http://www.njpw.co.jp/news/detail.php?nid=12708 |archive-date=November 12, 2014 }}&lt;/ref&gt;
     refusing to speak Japanese.&lt;ref name=&quot;slam2016&quot;&gt;{{cite web|url=http://slam.canoe.com/Slam/Wrestling/2016/06/18/22644039.html|title=Kenny Omega's stardom in Japan about far more than
     luck|last=Oliver|first=Greg|date=June 18, 2016|access-date=June 18, 2016|work=Slam! Sports|publisher=[[Canadian Online Explorer]]|archive-url=https://web.archive.org/web/20160618164023/http://
     slam.canoe.com/Slam/Wrestling/2016/06/18/22644039.html|archive-date=June 18, 2016|url-status=dead}}&lt;/ref&gt; In an interview, he stated that he was told that his otaku gimmick was &quot;too
     bubbly&quot; for Bullet Club, which led to him adopting the Cleaner nickname as a reference to [[Cleaner (crime)|people who clean up crime scenes]]. Smith cited the character [[Albert Wesker]] from
```



```
the ''[[Resident Evil]]'' video game series as well as [[Sylvester Stallone]]'s character, Marion &quot;Cobra&quot; Cobretti, from the film ''[[Cobra (1986 film)|Cobra]]'' as inspirations for the Cleaner gimmick.&lt;ref&gt;{{cite web|last1=Caro|url=https://www.fightful.com/wrestling/exclusives/golden-heartbreak-kenny-omega-and-art-long-term-storytelling|title=Golden Heartbreak: Kenny Omega And The Art of Long-term Storytelling|publisher=Fightful|date=December 5, 2020|access-date=April 26, 2021|archive-url=https://web.archive.org/web/20201206024933/https://www.fightful.com/wrestling/exclusives/golden-heartbreak-kenny-omega-and-art-long-term-storytelling|archive-date=December 6, 2020|url-status=live}}&lt;/ref&gt; Although Smith originally intended to embody the gimmick straightforwardly, he later integrated comedy into the persona as a response to people who thought he was portraying a janitor, doing so by coming out for his matches holding a mop and a broom.&lt;ref name=&quot;slam2016&quot;/&gt;
| ''[[The Expendables (2010 film)|The Expendables]]'' || &quot;Tool&quot; Ross || [[Sylvester Stallone]] ||
|Produced by [[Sylvester Stallone|Balboa Productions]]
''A Running Duck,'' also published as ''Fair Game,'' has been adapted twice into films; once as a [[Sylvester Stallone]] vehicle, ''[[Cobra (1986 film)|Cobra]]'',&lt;ref&gt;{{cite web|last=Temple |first=Robert |url=http://www.people.com/people/archive/article/0,,20093983,00.html |title=People Magazine |publisher=People.com |date=2015-01-28 |access-date=2016-01-28}}&lt;/ref&gt; and the second time as a film with [[Cindy Crawford]] entitled ''[[Fair Game (1995 film)|Fair Game]]''.&lt;ref&gt;{{cite web|url=https://movies.nytimes.com/person/309926/Paula-Gosling |archive-url=https://web.archive.org/web/20121104182854/http://movies.nytimes.com/person/309926/Paula-Gosling |url-status=dead |archive-date=2012-11-04 |title=New York Times |department=Movies &amp; TV Dept. |publisher=[[Baseline (database)|Baseline]] &amp; [[All Movie Guide]] |date=2012 |access-date=2016-01-28}}&lt;/ref&gt;
[[Category:Images of Sylvester Stallone]]</text>
      <text bytes="36782" sha1="mnt98eibr30b6x5lqf9zjzdbolqssh0" xml:space="preserve">{{Short description|Film by Sylvester Stallone}}
| director      = [[Sylvester Stallone]]
* Sylvester Stallone
* Sylvester Stallone
'''''Rambo''''' {{efn|Also known as '''''Rambo IV''''',&lt;ref name=&quot;former titles&quot;&gt;{{cite web|url=https://www.empireonline.com/movies/rambo/review/|title=Rambo Review|first=Roberto|last=Sadovski|work=Empire|date=February 1, 2007|access-date=August 28, 2018|url-status=live|archive-date=September 7, 2022|archive-url=https://web.archive.org/web/20220907173429/https://www.empireonline.com/movies/reviews/rambo-review/}}&lt;/ref&gt; or '''''Rambo: The Fight Continues'''''.&lt;ref name=&quot;dvdtitle&gt;{{cite web|url=https://www.forbes.com/sites/lukethompson/2019/09/02/blu-ray-picks-rambo-and-mib/#32147593e398|title=Blu-ray Picks: 'Rambo' And 'MIB'|first=Luke|last=Thompson|work=Forbes|date=September 2, 2019|access-date=September 27, 2019}}&lt;/ref&gt; The extended cut and select international versions are titled '''''John Rambo'''''.&lt;ref name=&quot;DVD Talk&quot;&gt;{{cite web|url=https://www.dvdtalk.com/reviews/43592/rambo/|title=Rambo - Extended Cut (Blu-ray)|first=Brian|last=Orndorf|work=DVD Talk|date=July 27, 2010|access-date=August 28, 2018}}&lt;/ref&gt;}} is a 2008 [[action film]] directed and co-written by [[Sylvester Stallone]], based on the character [[John Rambo]] created by author [[David Morrell]] for his novel ''[[First Blood (novel)|First Blood]]''.&lt;ref name=&quot;Rambo 2008&quot;&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/641887/Rambo/|title=Rambo 2008|work=[[Turner Classic Movies]]|access-date=March 24, 2016}}&lt;/ref&gt; A sequel to ''[[Rambo III]]'' (1988), it is the fourth installment in the [[Rambo (franchise)|''Rambo'' franchise]] and co-stars [[Julie Benz]], [[Paul Schulze]], [[Matthew Marsden]], [[Graham McTavish]], Rey Gallegos, [[Tim Kang]], [[Jake La Botz]], Maung Maung Khin, and [[Ken Howard]]. The film is dedicated to the memory of [[Richard Crenna]], who died in 2003. Crenna had played [[Sam Trautman|Colonel Sam Trautman]] in the previous films. In the film, Rambo (reprised by Stallone) leads a group of mercenaries into [[Myanmar|Burma]] to rescue Christian missionaries, who have been kidnapped by a local infantry unit.
* [[Sylvester Stallone]] as [[John Rambo]]
Stallone stated that due to the small production budget the only way to make the film memorable was to make it graphically violent. He said &quot;we were all sitting around in looking at the small production budget. Then I said  'Hey, fake blood is cheap, let's make it all-out bloody.'&quot; Filming started on January 22, 2007, and ended on May 4, 2007. It was shot in [[Chiang Mai, Thailand]] as well as in [[Mexico]] and the [[United States]] in [[Arizona]] and [[California]]. While filming near [[Burma]], Stallone and the rest of the crew narrowly avoided being shot by the Burmese military. Stallone described Burma as a &quot;hellhole&quot;. He said, &quot;We had shots fired above our heads&quot; and that he &quot;witnessed survivors with legs cut off and all kinds of land-mine injuries, maggot-infested wounds and ears cut off.&quot;&lt;ref name=&quot;NZ_Herald_10467584&quot;&gt;{{cite news |url=http://www.nzherald.co.nz/world/news/article.cfm?c_id=2&amp;objectid=10467584 |title=Sylvester Stallone describes Myanmar 'hellhole' |date=October 3, 2007 |agency=BANG Showbiz |work=[[The New Zealand Herald]] |access-date=November 4, 2011}}&lt;/ref&gt;
''Rambo'' received mixed reviews, with critics praising the film's action sequences and Stallone's performance, but criticizing the film's excessive violence.&lt;ref name=&quot;Huff&quot;/&gt; On [[Rotten Tomatoes]], the film has an approval rating of {{Rotten Tomatoes data|score}} based on {{Rotten Tomatoes data|count}} reviews, with an average rating of {{Rotten Tomatoes data|average}}. The site's critical consensus reads: &quot;Sylvester Stallone knows how to stage action sequences, but the movie's uneven pacing and excessive violence (even for the franchise) is more nauseating than entertaining.&quot;&lt;ref&gt;{{cite web |url=https://www.rottentomatoes.com/m/john_rambo/ |title=Rambo (Rambo IV)|access-date=March 24, 2020 |website=[[Rotten Tomatoes]]}}&lt;/ref&gt; On [[Metacritic]], the film has a weighted average score of 46 out of 100, based on 26 critics, indicating &quot;mixed or average reviews&quot;.&lt;ref&gt;{{cite web|website=[[Metacritic]]|title=Rambo|url=https://www.metacritic.com/movie/rambo}}&lt;/ref&gt; Audiences polled by [[CinemaScore]] gave the film an average grade of &quot;A-&quot; on an A+ to F scale.&lt;ref&gt;{{cite magazine |date=January 29, 2008 |author=Joshua Rich |title=''Spartans'' stops ''Rambo'' |url=https://ew.com/article/2008/01/29/spartans-stops-rambo/ |magazine=[[Entertainment Weekly]] |quote= Sylvester Stallone got a nice A- CinemaScore (its audience was mostly older males, duh) }}&lt;/ref&gt;&lt;ref name=&quot;CinemaScore&quot;&gt;{{cite web | url= https://www.cinemascore.com/publicsearch/index/title/ | title=CinemaScore |url-status= dead |archive-url= https://web.archive.org/web/20180206073531/https://www.cinemascore.com/publicsearch/index/title/ |archive-date= February 6, 2018 }}&lt;/ref&gt;
&lt;blockquote&gt;One film I'm truly proud of - it's the best action film I've ever done because it's the most truthful - is ''Rambo IV'', dealing with Burma, where they've had a civil war for 67 years. But I got excoriated because the movie's so violent. And it is violent. It's horrifying. It's children being burnt alive. That's what makes civil war worse than anything: It's your neighbor, all of a sudden, killing you. I was really happy with that film, and I never thought it would ever reach the theater. I thought, "They're never going to show this.&lt;ref&gt;{{cite web | url=https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/ | title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot; | website=[[The Hollywood Reporter]] | date=November 7, 2022 }}&lt;/ref&gt;&lt;/blockquote&gt;
In 2009, Stallone announced plans for a fifth film titled ''Rambo V: The Savage Hunt''. The film would have been loosely based on ''[[Hunter (Huggins novel)|Hunter]]'' by James Byron Huggins and would have focused on Rambo leading an elite special forces kill team to hunt and kill a genetically engineered creature.&lt;ref&gt;{{cite web|url=https://www.slashfilm.com/more-details-on-rambo-5-the-savage-hunt-creature-feature-based-on-the-sci-fi-book-hunter/|title=More Details on Rambo 5: The Savage Hunt - Creature Feature Based on the Sci-Fi Book Hunter|first=Hunter|last=Stephenson|work=SlashFilm|date=September 6, 2009|access-date=October 2, 2018}}&lt;/ref&gt; In 2011, Sean Hood was hired to write a new script, separate from ''The Savage Hunt'', titled ''Rambo: Last Stand'' that Hood described was &quot;more in line with the small-town thriller of ''[[First Blood]]''&quot;.&lt;ref&gt;{{cite web|url=https://www.empireonline.com/movies/news/new-screenplay-rambo-5/|title=New Screenplay For Rambo 5|first=Owen|last=Williams|work=Empire|date=August 19, 2011|access-date=October 2, 2018}}&lt;/ref&gt; In 2012, Hood revealed that ''Rambo V'' was on hold while Stallone finishes ''[[The Expendables 2]]'' (2012). Hood also revealed his uncertainty whether the film will be similar to ''Unforgiven'' or will be a passing-of-the-torch.&lt;ref&gt;{{cite web|url=https://www.joblo.com/horror-movies/news/sean-hood-finally-gives-an-update-on-the-state-of-rambo-5|title=Sean Hood Finally Gives an Update on the State of Rambo 5|first=Marcey|last=Papandrea|work=Joblo|date=February 10, 2012|access-date=October 2, 2018}}&lt;/ref&gt; In 2016, Sylvester Stallone revealed that ''Rambo V'' was no longer in production.&lt;ref&gt;{{cite web|work=[[Variety (magazine)|Variety]]|url=https://variety.com/2016/film/news/sylvester-stallone-retiring-rambo-1201672291/|title=Sylvester Stallone Is Retiring From Playing Rambo|author=Setoodeh, Ramin|date=January 5, 2016}}&lt;/ref&gt;
In May 2018, ''Rambo V'' was re-announced and was scheduled to begin filming in September with the plot focusing on Rambo taking on a Mexican drug cartel.&lt;ref name=&quot;ScreenDaily&quot;&gt;{{cite magazine|last=Kay|first=Jeremy|url=https://www.screendaily.com/sylvester-stallone-lining-up-rambo-v-exclusive/5128882.article|title=Sylvester Stallone lining up 'Rambo V' (exclusive)|magazine=[[Screen Daily]]|date=May 5, 2018|access-date=May 8, 2018}}&lt;/ref&gt; Stallone was confirmed to be co-writing the script with Matt Cirulnick, but was unlikely to direct.&lt;ref&gt;{{cite web|last=Wiseman|first=Andrew|work=[[Deadline Hollywood|Deadline]]|url=https://deadline.com/2018/05/sylvester-stallone-in-line-to-take-on-mexican-cartel-in-rambo-5-1202383468/|title=Sylvester Stallone In Line To Take on Mexican Cartel in Rambo 5|date=May 5, 2018|access-date=May 7, 2018}}&lt;/ref&gt; That same month, Stallone confirmed that the film was scheduled for a fall 2019 release.&lt;ref name=&quot;IGN&quot;&gt;{{cite news|last=Gilyadov|first=Alex|url=https://www.ign.com/articles/2018/05/08/rambo-5-reportedly-in-the-works-with-sylvester-stallone|title=Rambo 5 Confirmed by Stallone, Coming Fall 2019|website=IGN|date=May 8, 2018|access-date=May 8, 2018}}&lt;/ref&gt; In August 2018, [[Adrian Grunberg]] was announced as the director.&lt;ref&gt;{{cite web|url=https://screenrant.com/rambo-5-director/|title=Rambo 5 Reportedly Recruits Get the Gringo Director|first=David|last=Stephens|work=Screen Rant|date=August 18, 2018|access-date=October 2, 2018}}&lt;/ref&gt; Principal photography began in October 2018.&lt;ref&gt;{{cite web|url=https://collider.com/rambo-5-filming-details-sylvester-stallone/|title=Sylvester Stallone Is a Cowboy for Some Reason as 'Rambo 5' Starts Filming|first=Dave|last=Trumbore|work=Collider|date=October 2, 2018|access-date=October 2, 2018}}&lt;/ref&gt; ''Rambo: Last Blood'' was theatrically released in the United States on September 20, 2019.
{{Sylvester Stallone}}
[[Category:Films directed by Sylvester Stallone]]
[[Category:Films with screenplays by Sylvester Stallone]]
* ''[[Spy Kids:Game Over|Spy Kids 3D:Game Over]]'' Starring [[Alexa Vega]],[[Sylvester Stallone]],[[Emily Osment]] &amp; [[Antonio Banderas]]
Artists involved include: [[Will Ferrell]], [[Sarah Silverman]], [[Jennifer Aniston]], [[Woody Harrelson]], [[James Cameron]], [[Judd Apatow]], [[William Baldwin]], [[Hank Azaria]], [[Michelle Krusiec]], [[Tila Tequila]], [[Kim Kardashian]], [[Damian Marley]], [[Sheryl Crow]], [[Felicity Huffman]], [[Elliot Page]], [[Joseph Fiennes]], [[Jason Schwartzman]], [[Eddie Izzard]], [[Jorja Fox]], [[Eric Szmanda]], [[Anjelica Huston]], [[Famke Janssen]], [[Sylvester Stallone]], [[Steven Seagal]], [[Norman Lear]], [http://www.plumvillage.org Tich Naht Hahn], [[Brett Dennen]], [[Matisyahu]], [[Giovanni Ribisi]], [[Diego Maradona]], [[Maná|Mana]], [[Julie Benz]], [[Eva Longoria]],  [[Davood Roostaei]], [[Jackson Browne]], [[Wallace Langham]], [[Jason Biggs]], and [[Jenny Mollen]].
* [[Sylvester Stallone]]&lt;ref name=&quot;Bathurst 2005&quot;&gt;{{cite web | last=Bathurst | first=Bella | title=Focus: How I learnt to love my silent world | website=The Guardian | date=23 January 2005 | url=http://www.theguardian.com/society/2005/jan/23/health.lifeandhealth | access-date=1 March 2023}}&lt;/ref&gt;
===Split with Sylvester Staline===
| name       = Fuck the Facts / Sylvester Staline
Keller began his screenwriting career by writing [[Marc Forster]]'s film ''[[Machine Gun Preacher]]'', which  was released by [[Relativity Media]] in 2011.&lt;ref&gt;{{cite web|url=http://blogs.indiewire.com/theplaylist/jason_keller_signs_on_to_write_the_passage_for_matt_reeves|title=Jason Keller Signs On To Write Vampire Pic 'The Passage' For Matt Reeves|
```



```
publisher=IndieWire|first=Matthew|last=Newlin|date=June 21, 2011|access-date=January 10, 2014|archive-date=January 10, 2014|archive-url=https://web.archive.org/web/20140110151401/http://blogs.indiewire.com/theplaylist/jason_keller_signs_on_to_write_the_passage_for_matt_reeves|url-status=dead}}&lt;/ref&gt; Keller went on to write [[Tarsem Singh]]'s film ''[[Mirror Mirror (film)|Mirror Mirror]]'', starring [[Julia Roberts]],&lt;ref&gt;{{cite web|url=http://www.hollywoodreporter.com/news/relativitys-snow-white-movie-be-257773|title=Relativity's Snow White Movie to Be Called 'Mirror Mirror'|publisher=The Hollywood Reporter|first=Philiana|last=Ng|date=November 4, 2011}}&lt;/ref&gt; as well as [[Mikael Håfström]]'s 2013 action film ''[[Escape Plan (film)|Escape Plan]]'', starring [[Arnold Schwarzenegger]] and [[Sylvester Stallone]].&lt;ref name = PenName/&gt; He is also the executive producer of the fifth movie in the ''[[Die Hard]]'' franchise, ''[[A Good Day to Die Hard]]'', starring [[Bruce Willis]].&lt;ref&gt;{{cite web|url=http://www.screendaily.com/reviews/the-latest/a-good-day-to-die-hard/5051955.article|title=A Good Day To Die Hard|publisher=Screen International|first=Tim|last=Grierson|date=February 13, 2013}}&lt;/ref&gt;
1952 *''[[Creed II]]'': [[Rocky Balboa]] and [[Donnie Creed]] (Sylvester Stallone and Michael B. Jordan) hold their training camp in the desert of Deming.&lt;ref&gt;{{cite news |last=Cortes Gonzalez |first=Maria |date=2018-06-14 |title=Sylvester Stallone and Michael B. Jordan spotted in Deming for 'Creed II' |url=https://www.elpasotimes.com/story/entertainment/2018/06/14/sylvester-stallone-rocky-michael-jordan-film-creed-ii-deming/701703002/ |work=El Paso Times |location= |access-date=2024-01-14}}&lt;/ref&gt;
1953 He was hired by [[Carolco Pictures]] to direct ''[[Rambo III]]'' (1988), but was replaced with second-unit director [[Peter MacDonald (director)|Peter MacDonald]]. The collaboration had started well but it became apparent that he had major creative differences with star/writer [[Sylvester Stallone]] so Mulcahy left the project. &quot;It was nobody's fault,&quot; said Mulcahy later. &quot;Sly is a wonderful man, and we still remain friends.&quot;&lt;ref name=&quot;two&quot;/&gt;&lt;ref&gt;{{Cite web|title=Rambo III|url=https://catalog.afi.com/Film/59082-RAMBO-III?sid=04fe72b4-cb46-4c73-a8b8-fed17728627b&amp;sr=10.999381&amp;cp=1&amp;pos=0|access-date=2021-06-11|website=catalog.afi.com}}&lt;/ref&gt;
1954 | holder        = [[Sylvester Stallone]],&lt;br /&gt;''[[Expend4bles]]'' (2023)
1955 | [[Sylvester Stallone]]
1956 | [[Sylvester Stallone]]
1957 | '''[[Sylvester Stallone]]'''
1958 |'''[[Sylvester Stallone]]'''
1959 *[[Sylvester Stallone]]
1960 * [[Sylvester Stallone]]
1961 In 1997 [[James Mangold]]'s ''[[Cop Land]]'' was well-received critically with a cast that included [[Sylvester Stallone]], [[Robert De Niro]], [[Ray Liotta]], and [[Harvey Keitel]]. Through Independent Pictures, Woods produced films for [[New Line Cinema]]'s division, [[Fine Line Features]], such as [[Harmony Korine]]'s directorial debut, ''[[Gummo]]''.
1962 '''Alain Dorval''' (born '''Alain Fernand Jean-Marie Bergé''', 9 August 1946 – 13 February 2024) was a French actor, best known as the official dub voice of [[Sylvester Stallone]], [[Nick Nolte]], and the [[The Walt Disney Company|Disney]] character [[Pete (Disney)|Pete]]. His voice was also featured on the radio stations [[Skyrock (radio)|Skyrock]] and [[ADO FM]].
1963 Dorval was educated at the [[Cours Simon]] drama school and the [[Conservatory of Dramatic Art of Paris]].&lt;ref name=&quot;parisien&quot;&gt;{{cite web |title=Alain Dorval, voix française de Sylvester Stallone et père de la ministre Aurore Bergé, est mort à 77 ans |url=https://www.leparisien.fr/culture-loisirs/cinema/alain-dorval-voix-francaise-de-sylvester-stallone-et-pere-de-la-ministre-aurore-berge-est-mort-a-77-ans-13-02-2024-LBRHLKLMV5EZXE4PCFFCEHLJYY.php |website=[[Le Parisien]] |access-date=16 February 2024 |language=French |date=13 February 2024}}&lt;/ref&gt; His daughter is the [[Renaissance (French political party)|Renaissance]] politician [[Aurore Bergé]];&lt;ref name=&quot;parisien&quot; /&gt; during his daughter's marriage, he was the father-in-law of politician [[Nicolas Bays]].&lt;ref&gt;Nolwenn Le Blevennec (18 November 2016), [https://www.nouvelobs.com/rue89/rue89-politique/20131028.RUE9655/aurore-berge-26-ans-la-candidate-ump-qui-a-tout-pour-plaire-a-la-gauche.html Aurore Bergé, 26 ans, la candidate UMP qui a tout pour plaire à la gauche]&amp;nbsp;''[[L'Obs]]''.&lt;/ref&gt;
1964 He died from cancer at the [[Institut Gustave Roussy]] in [[Villejuif]] on 13 February 2024, at the age of 77.&lt;ref name=&quot;parisien&quot; /&gt;&lt;ref&gt;{{Cite web |date=13 February 2024 |title=Mort du comédien Alain Dorval, la voix française de Sylvester Stallone |url=https://www.francetvinfo.fr/culture/cinema/mort-du-comedien-alain-dorval-la-voix-francaise-de-sylvester-stallone_6363346.html |access-date=13 February 2024 |website=Franceinfo |language=fr-FR}}&lt;/ref&gt;&lt;ref&gt;[https://www.libramemoria.com/deces-celebres/2024/02/13/le-doubleur-francais-alain-dorval-est-decede Le doubleur français Alain Dorval est décédé] {{in lang|fr}}&lt;/ref&gt;
1965 The album saw Azúcar Moreno making their first collaborations with some of the top names in the world of Latin music in the United States, such as former [[Miami Sound Machine]] member [[Emilio Estefan Jr.]], composer and producer Estéfano, [[Kike Santander]] and Juan R. Marquez, the team behind numerous hits for world-renowned artists such as [[Julio Iglesias]], [[Gloria Estefan]], [[Jon Secada]], [[Shakira]] and [[Chayanne]]. The title track and the biggest hit &quot;El Amor&quot; was featured in the Hollywood movie ''[[The Specialist]]'', starring [[Sylvester Stallone]] and [[Sharon Stone]]. The extended 12&quot; mix of the song was included on the CD version of ''El Amor'' and a special edition of the soundtrack album called ''The Specialist - The Remixes'' featured [[Tony Moran]]'s twelve minutes forty-two seconds ''Multiple Orgasm Mix'', and the song managed to cross over to the US dance charts, reaching #50 on ''[[Billboard (magazine)|Billboard]]'''s [[Hot Dance Club Play]] listing in early 1995.&lt;ref&gt;[{{AllMusic|class=album|id=r271080|pure_url=yes}} All Music ''El Amor'']&lt;/ref&gt; The follow-up single &quot;No Será Facil&quot;, produced by the same team, was written by Gloria Estefan and originally recorded by the Miami Sound Machine on their 1982 album ''Rio''. Further hits from the album include &quot;No Tendré Piedad&quot; and &quot;Hay Que Saber Perder&quot;.
1966 * [[Sylvester Stallone]] jogging to the melody of &quot;[[Gonna Fly Now]]&quot; in ''[[Rocky]]'' (1976).
1967 * [[Sylvester Stallone]] shooting around as [[John Rambo]] in ''[[Rambo: First Blood Part II]]'' (1985).
1968 *[[Sylvester Stallone]]
1969 The film ''[[Rambo (2008 film)|Rambo]]'' was released worldwide in early 2008, with [[Sylvester Stallone]] continuing his role as the [[John Rambo|eponymous hero]]. In it, a fictionalised Burmese military played the role of the 'evil oppressors' and, although the film didn't make it to Burmese cinema screens, it became a huge underground success amongst the Burmese population.&lt;ref&gt;{{cite web|url=http://www.bangkokpost.com/topstories/topstories.php?id=125671|title=Akkharaphongpricha name withdrawn - Bangkok Post: topstories|author=Post Publishing PCL.|publisher=|accessdate=29 November 2014|archive-date=4 December 2014|archive-url=https://web.archive.org/web/20141204141118/http://www.bangkokpost.com/topstories/topstories.php?id=125671|url-status=live}}&lt;/ref&gt; Research for the movie was obtained, in large part, from FBR field reports.&lt;ref&gt;{{cite web|url=http://www.coventrytelegraph.net/whats-on-coventry-warwickshire/cinema-film/2008/02/22/sly-muscles-back-in-on-the-rambo-franchise-92746-20511428/|title=Coventry Telegraph: Coventry news, sport, lifestyle and events|work=Coventry Telegraph|accessdate=29 November 2014|archive-date=10 February 2012|archive-url=https://web.archive.org/web/20120210023817/http://www.coventrytelegraph.net/whats-on-coventry-warwickshire/cinema-film/2008/02/22/sly-muscles-back-in-on-the-rambo-franchise-92746-20511428/|url-status=dead}}&lt;/ref&gt;
1970 | starring       = [[Sylvester Stallone]]&lt;br&gt;Antony Page&lt;br&gt;Rebecca Grimes&lt;br&gt;Roy White&lt;br&gt;Vickie Lancaster&lt;br&gt;Dennis Tate&lt;br&gt;Barbara Lee Govan&lt;br&gt;
1971 '''''No Place to Hide''''' is a 1973 American [[thriller film]] directed by Robert Allen Schnitzer and starring [[Sylvester Stallone]]. The film is about New York in the late 1960s; a politically motivated group of students plans bombings of company offices who do business with dictators in Central American countries. But when they contact a known terrorist and bombing specialist, the [[Federal Bureau of Investigation|FBI]] gets on their track. The film was re-cut and re-released as '''''Rebel''''' with greater emphasis on Stallone's character after he rose to fame.  In 1990, the film was again re-edited, this time as a comedy, and released under the title '''''A Man Called... Rainbo'''''.
1972 * [[Sylvester Stallone]] as Jerry Savage
1973 Because of numerous production problems, the [[independent film|independent project]] shoot was moved from Miami to Mexico.&lt;ref name=&quot;McKittrick&quot;&gt;{{cite web |date=11 February 2017 |author=Christopher McKittrick |url=http://lwlies.com/articles/the-good-life-sylvester-stallone-dennis-hopper/ |title=The wild, untold story of The Good Life |publisher=Little White Lies |access-date=1 February 2021 }}&lt;/ref&gt; The film was never released commercially due to legal action initiated by [[Sylvester Stallone]], who did a cameo in the film as a favor to his brother. He claims that a promotion reel used his scene excessively, therefore was promoting him as the star, instead of the cameo that he had agreed upon. He sued for $20 million; his average salary at the time. The producers countersued, and, in 1999, the case was settled out of court.&lt;ref name=&quot;McKittrick&quot; /&gt;
1974 * [[Sylvester Stallone]] had a [[Cameo appearance|cameo]] as a Boss
1975 The presence of Sylvester Stallone in the film was limited to a simple cameo, a favor he, along with his brother Frank, the protagonist of the film, made to the brothers Alan and Diane Mehrez, producers of the film. The problems arose when the Mehrez brothers promoted the film as soon as the shooting was over, as if Stallone were the protagonist; the plaintiff sued the two producers and the distribution of the film was thus blocked. The film was canceled and still remains unpublished, as Stallone asked for 20 million dollars to let the film be distributed with its leading name, his average salary during that period.
1976 *[[Sylvester Stallone]] (actor)
1977 Notable members of the club have included [[Sean Connery]], [[Kenny G]], [[Mel Gibson]], [[Caitlyn Jenner]], [[Wayne Gretzky]], [[Scott Hamilton (figure skater)|Scott Hamilton]], [[Kevin James]], [[Tom Kelly (musician)|Tom Kelly]], [[Angelo Mozilo]], [[Craig T. Nelson]], [[Jack Nicholson]], [[Joe Pesci]], [[Tom Selleck]], [[Jane Seymour (actress)|Jane Seymour]], [[O. J. Simpson]], [[Will Smith]], [[Kevin Sorbo]], [[Sylvester Stallone]],&lt;ref&gt;{{cite journal|last=Monday|first=Glenn|title=Tales from the 19th Hole - Sylvester Stallone|journal=Southland Golf Magazine|date=March 2012}}&lt;/ref&gt; and [[Justin Timberlake]].&lt;ref&gt;{{cite news|last=Baum|first=Gary|title=L.A.'s Power Golf Clubs: Where the Hollywood Elite Play|newspaper=The Hollywood Reporter|date=23 June 2011}}&lt;/ref&gt;
1978 * [[Driven (2001 film)|''Driven'' (2001 film)]], an American film starring Sylvester Stallone
1979 '''Thomas Danneberg''' (2 June 1942 – 30 September 2023) was a German actor. He is best known as Charles Emerson in the 1967 [[Films based on Edgar Wallace works|Edgar Wallace film]] ''The Blue Hand'' and for dubbing over the actors [[Arnold Schwarzenegger]], [[Terence Hill]], [[Sylvester Stallone]], [[John Cleese]], [[Dan Aykroyd]], [[Adriano Celentano]], [[Nick Nolte]], [[John Travolta]], [[Michael York (actor)|Michael York]], [[Rutger Hauer]] and [[Dennis Quaid]].
1980 *[[Sylvester Stallone]]
1981 # 2008-09-22 08:49:07 (UTC): User [[:az:User:87.120.50.100|az:87.120.50.100]] ([[:az:User talk:87.120.50.100|talk]] - [[:az:Special:Contributions/87.120.50.100|contribs]]; 10) to [[:az:Silvestr Stallone]] ([http://az.wikipedia.org/w/index.php?title=Silvestr_Stallone&amp;diff=327538&amp;oldid=320559 diff] - [http://az.wikipedia.org/w/index.php?title=Silvestr_Stallone&amp;action=edit&amp;undoafter=320559&amp;undo=327538 undo]) - Link: www.stallone-the-best.piczo.com/.&lt;!--ID 11607365--&gt;&lt;br/&gt;* Links added in this diff: www.stallone-the-best.piczo.com/ (10, 10, 10, 7)
1982 # 2008-09-22 08:52:28 (UTC): User [[:bs:User:87.120.50.100|bs:87.120.50.100]] ([[:bs:User talk:87.120.50.100|talk]] - [[:bs:Special:Contributions/87.120.50.100|contribs]]; 10) to [[:bs:Sylvester Stallone]] ([http://bs.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=748848&amp;oldid=740502 diff] - [http://bs.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;action=edit&amp;undoafter=740502&amp;undo=748848 undo]) - Link: www.stallone-the-best.piczo.com/.&lt;!--ID 11607395--&gt;&lt;br/&gt;* Links added in this diff: www.stallone-the-best.piczo.com/ (10, 10, 10, 7)
```



1983 # 2008-09-22 09:00:03 (UTC): User [[:es:User:87.120.50.100|es:87.120.50.100]] ([[:es:User talk:87.120.50.100|talk]] - [[:es:Special:Contributions/87.120.50.100|contribs]]; 10) to [[:es:Sylvester Stallone]] ([http://es.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=20353113&amp;oldid=20154742 diff] - [http://es.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;action=edit&amp;undoafter=20154742&amp;undo=20353113 undo]) - Link: www.stallone-the-best.piczo.com/.&lt;!--ID 11607464--&gt;&lt;br/&gt;* Links added in this diff: www.stallone-the-best.piczo.com/ (10, 10, 10, 7)

1984 # 2008-09-22 12:27:08 (UTC): User [[:fi:User:87.120.50.100|fi:87.120.50.100]] ([[:fi:User talk:87.120.50.100|talk]] - [[:fi:Special:Contributions/87.120.50.100|contribs]]; 10) to [[:fi:Sylvester Stallone]] ([http://fi.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=5511975&amp;oldid=5412169&amp;rcid=5645968 diff] - [http://fi.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;action=edit&amp;undoafter=5412169&amp;undo=5511975 undo]) - Link: www.stallone-the-best.piczo.com/.&lt;!--ID 11609604--&gt;&lt;br/&gt;* Links added in this diff: www.stallone-the-best.piczo.com/ (10, 10, 10, 7)

1985 # 2008-09-28 11:07:36 (UTC): User [[User:87.120.50.100|87.120.50.100]] ([[User talk:87.120.50.100|talk]] - [[Special:Contributions/87.120.50.100|contribs]]; 10) to [[Sylvester Stallone]] ([http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=241511066&amp;oldid=241394759 diff] - [http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;action=edit&amp;undoafter=241394759&amp;undo=241511066 undo]) - Link: www.stallone-the-best.piczo.com/.&lt;!--ID 11929803--&gt;&lt;br/&gt;* Links added in this diff: www.stallone-the-best.piczo.com/ (10, 10, 10, 7)

1986 # 2008-09-24 13:32:39 (UTC): User [[:pt:User:189.18.71.24|pt:189.18.71.24]] ([[:pt:User talk:189.18.71.24|talk]] - [[:pt:Special:Contributions/189.18.71.24|contribs]]; 5) to [[:pt:Sylvester Stallone]] ([http://pt.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=12431269&amp;oldid=12415936&amp;rcid=13032673 diff] - [http://pt.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;action=edit&amp;undoafter=12415936&amp;undo=12431269 undo]) - Link: www.sylvesterstallonebrasil.webs.com.&lt;!--ID 11721300--&gt;&lt;br/&gt;* Links added in this diff: www.sylvesterstallonebrasil.webs.com (5, 4, 4, 3)

1987 # 2008-09-24 13:36:51 (UTC): User [[:pt:User:189.18.71.24|pt:189.18.71.24]] ([[:pt:User talk:189.18.71.24|talk]] - [[:pt:Special:Contributions/189.18.71.24|contribs]]; 5) to [[:pt:Sylvester Stallone]] ([http://pt.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=12431342&amp;oldid=12431312&amp;rcid=13032749 diff] - [http://pt.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;action=edit&amp;undoafter=12431312&amp;undo=12431342 undo]) - Link: www.sylvesterstallonebrasil.webs.com/.&lt;!--ID 11721566--&gt;&lt;br/&gt;* Links added in this diff: www.sylvesterstallonebrasil.webs.com/ (5, 4, 4, 3)

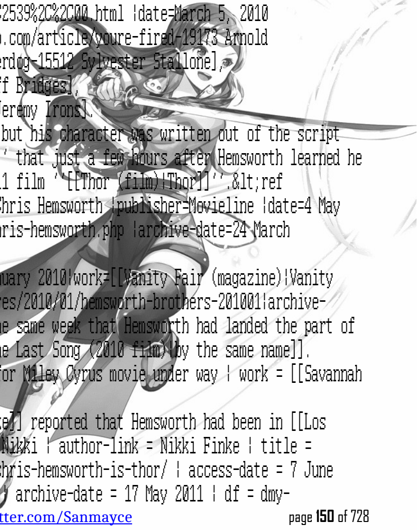
1988 # 2008-09-24 13:38:51 (UTC): User [[User:189.18.71.24|189.18.71.24]] ([[User talk:189.18.71.24|talk]] - [[Special:Contributions/189.18.71.24|contribs]]; 5) to [[Sylvester Stallone]] ([http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=240678213&amp;oldid=240654308 diff] - [http://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;action=edit&amp;undoafter=240654308&amp;undo=240678213 undo]) - Link: www.sylvesterstallonebrasil.webs.com/.&lt;!--ID 11721809--&gt;&lt;br/&gt;* Links added in this diff: www.sylvesterstallonebrasil.webs.com/ (5, 4, 4, 3)

1989 # 2008-09-24 13:42:30 (UTC): User [[:fr:User:189.18.71.24|fr:189.18.71.24]] ([[:fr:User talk:189.18.71.24|talk]] - [[:fr:Special:Contributions/189.18.71.24|contribs]]; 5) to [[:fr:Sylvester Stallone]] ([http://fr.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=33721071&amp;oldid=33676900&amp;rcid=33925886 diff] - [http://fr.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;action=edit&amp;undoafter=33676900&amp;undo=33721071 undo]) - Link: www.sylvesterstallonebrasil.webs.com/.&lt;!--ID 11722050--&gt;&lt;br/&gt;* Links added in this diff: www.sylvesterstallonebrasil.webs.com/ (5, 4, 4, 3)

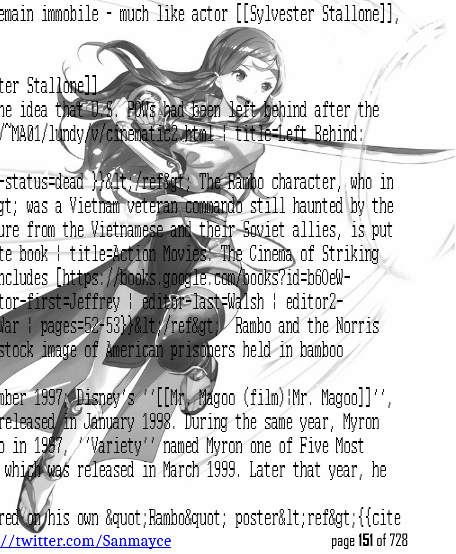
1990 The film screened at 67th [[Venice Film Festival]].&lt;ref&gt;{{cite news |url=http://articles.timesofindia.indiatimes.com/2010-09-25/news-interviews/28220422_1_mani-ratnam-filmmaker-award-suhasini |archive-url=https://archive.today/20120711025851/http://articles.timesofindia.indiatimes.com/2010-09-25/news-interviews/28220422_1_mani-ratnam-filmmaker-award-suhasini |url-status=dead |archive-date=11 July 2012 |title=Raavanan enthralls Venice |author=Asish Samanta |access-date=25 September 2010 |newspaper=[[The Times of India]] |date=25 September 2010}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.rediff.com/movies/report/showcasing-raavan-in-venice/20100907.htm |title=Showcasing Raavan in Venice |publisher=Rediff |date=7 September 2010 |access-date=30 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131101025128/http://www.rediff.com/movies/report/showcasing-raavan-in-venice/20100907.htm |archive-date= 1 November 2013 }}&lt;/ref&gt; The film was well received at [[Venice]] where the audience gave it a thunderous applause after it was screened.&lt;ref&gt;{{cite web|url=http://www.bollywoodtrade.com/trade-news/raavan-and-raavanan-a-big-hit-in-film-festivals/1935.htm |title=Raavan and Raavanan a big hit in film festivals |publisher=Bollywood Trade |date=27 September 2010 |access-date=30 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131031212354/http://www.bollywoodtrade.com/trade-news/raavan-and-raavanan-a-big-hit-in-film-festivals/1935.htm |archive-date=31 October 2013 }}&lt;/ref&gt; The master filmmaker was honoured with the [[Jaeger-LeCoultre]] Glory to the Filmmaker Award, an award shared by the likes of legends like Takeshi Kitano, Abbas Kiarostami and Sylvester Stallone.&lt;ref&gt;{{cite news |url=http://articles.timesofindia.indiatimes.com/2010-10-06/news-interviews/28244828_1_raavanan-filmmaker-award-suhasini-and-vikram |archive-url=https://web.archive.org/web/20120503103152/http://articles.timesofindia.indiatimes.com/2010-10-06/news-interviews/28244828_1_raavanan-filmmaker-award-suhasini-and-vikram |url-status=dead |archive-date=3 May 2012 |title=Raavanan goes to Korea |author=P Sangeetha |access-date=6 October 2010 |newspaper=[[The Times of India]] |date=6 October 2010}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.rediff.com/movies/report/mani-ratnam-honoured-at-venice-film-festival/20100908.htm |title=Mani Ratnam honoured at Venice Film Festival |publisher=Rediff |date=8 September 2010 |access-date=30 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131101031101/http://www.rediff.com/movies/report/mani-ratnam-honoured-at-venice-film-festival/20100908.htm |archive-date= 1 November 2013 }}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.rediff.com/movies/slide-show/slide-show-1-abhi-ash-cheer-for-mani-ratnam/20100518.htm |title=Ash, Abhishek cheer for Mani Ratnam in Cannes |publisher=Rediff |date=18 May 2010 |access-date=30 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131101025659/http://www.rediff.com/movies/slide-show/slide-show-1-abhi-ash-cheer-for-mani-ratnam/20100518.htm |archive-date= 1 November 2013 }}&lt;/ref&gt; Later, the film has been screened at the [[15th Busan International Film Festival]].&lt;ref&gt;{{cite news|url=http://www.thehindu.com/features/cinema/raavanan-at-pusan-festival/article819816.ece |title=Raavanan at Pusan festival |newspaper=The Hindu |date=8 October 2010 |access-date=27 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131029213915/http://www.thehindu.com/features/cinema/raavanan-at-pusan-festival/article819816.ece |archive-date=29 October 2013 }}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=http://www.hindu.com/fr/2010/11/05/stories/2010110550730100.htm |title=French star steals the show |author=Lekha J. Shankar |date=5 November 2010 |access-date=9 October 2013 |url-status=dead |archive-url=https://web.archive.org/web/20131029222121/http://www.hindu.com/fr/2010/11/05/stories/2010110550730100.htm |newspaper=[[The Hindu]] |archive-date=29 October 2013 }}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=http://movies.ndtv.com/regional/raavanan-at-pusan-international-film-festival-58042 |title=Raavanan at Pusan International Film Festival |publisher=NDTV |agency=Press Trust of India |date=8 October 2010 |access-date=27 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131108235325/http://movies.ndtv.com/regional/raavanan-at-pusan-international-film-festival-58042 |archive-date= 8 November 2013 }}&lt;/ref&gt; Indian Panorama Film Festival.&lt;ref&gt;{{cite news|url=http://www.hindu.com/fr/2011/03/04/stories/2011030450260400.htm |title=Port calling: Indian Panorama Film Festival |author=Ziya Us Salam |date=4 March 2011 |access-date=9 October 2013 |url-status=dead |archive-url=https://web.archive.org/web/20131029221431/http://www.hindu.com/fr/2011/03/04/stories/2011030450260400.htm |newspaper=[[The Hindu]] |archive-date=29 October 2013 }}&lt;/ref&gt; The film was premiered at the 10th Annual Mahindra [[Indo-American Arts Council]] Film Festival in New York City.&lt;ref&gt;{{cite web|url=http://www.thaindian.com/newsportal/entertainment/raavanan-will-be-screened-at-miaac-2010_100446882.html |title=Raavanan Will Be Screened at MIAAC 2010 |publisher=Thaindian |author=Sampurn Wire |date=20 October 2010 |access-date=27 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131029210129/http://www.thaindian.com/newsportal/entertainment/raavanan-will-be-screened-at-miaac-2010_100446882.html |archive-date=29 October 2013 }}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.iaac.us/MIAAC2010/raavanan/ |title=Tenth Annual MIAAC Film Festival |work=iaac |access-date=27 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131029211044/http://www.iaac.us/MIAAC2010/raavanan/ |archive-date=29 October 2013 }} Tenth Annual MIAAC Film Festival&lt;/ref&gt; ''Jim Luce'' praised mani ratnam's work and mentioned 'Raavanan is a must see international film of 2010.&lt;ref&gt;{{cite news|url=https://www.huffingtonpost.com/jim-luce/mani-ratnams-raavanan-the_b_784529.html |title=Mani Ratnam's Raavanan: The Must-See International Film of 2010 |newspaper=HuffPost |author=Jim Luce |date=16 November 2010 |access-date=16 November 2010 |url-status=live |archive-url=https://web.archive.org/web/20101119035936/http://www.huffingtonpost.com/jim-luce/mani-ratnams-raavanan-the_b_784529.html |archive-date=19 November 2010 }}&lt;/ref&gt; 29th San Francisco Asian American Film Festival.&lt;ref&gt;{{cite web|url=http://behindwoods.com/tamil-movie-news-1/feb-11-03/mani-rathnam-raavanan-19-02-11.html |title=Raavanan still a favorite |website=Behindwoods |date=19 February 2011 |access-date=30 October 2013 |url-status=live |archive-url=https://web.archive.org/web/20131101042820/http://behindwoods.com/tamil-movie-news-1/feb-11-03/mani-rathnam-raavanan-19-02-11.html |archive-date= 1 November 2013 }}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://caam.gala-engine.com/2011/films-events/program/raavanan/ |title=San Francisco International Asian American Film Festival |publisher=[[Center for Asian American Media]] |access-date=30 October 2013 |url-status=dead |archive-url=https://web.archive.org/web/20131101094453/http://caam.gala-engine.com/2011/films-events/program/raavanan/ |archive-date=1 November 2013 }}&lt;/ref&gt;

1991 As a contributing editor for [[Cigar Aficionado]] magazine, he has written more than 40 cover stories since 2009.&lt;ref&gt;[http://www.cigaraficionado.com/Cigar/CA_Profiles/People_Index_Page/0,2539,,00.html] {{webarchive|url=https://web.archive.org/web/20100305051815/http://www.cigaraficionado.com/Cigar/CA_Profiles/People_Index_Page/0%2C2539%2C%2C00.html |date=March 5, 2010 }}&lt;/ref&gt; Among his cover story subjects were [https://www.cigaraficionado.com/article/the-music-mogul-6238 Jay-Z], [https://www.cigaraficionado.com/article/youre-fired-19173 Arnold Schwarzenegger], [https://www.cigaraficionado.com/article/the-paisley-patterns-16421 Brad Paisley], [https://www.cigaraficionado.com/article/the-underdog-15512 Sylvester Stallone], [https://www.cigaraficionado.com/article/the-unstoppable-force-18773 Sean Combs], [https://www.cigaraficionado.com/article/living-the-dream-17139 Jeff Bridges], [https://www.cigaraficionado.com/article/an-appetite-for-life-17224 Stanley Tucci] and [https://www.cigaraficionado.com/article/smoking-irons-16970 Jeremy Irons].

1992 In 2009, Hemsworth was selected to act opposite [[Sylvester Stallone]] in Stallone's 2010 film, ''[[The Expendables (2010 film)|The Expendables]]'', but his character was written out of the script (Hemsworth later co-starred in ''[[The Expendables 2]]'').&lt;ref name=&quot;chrisverge&quot; /&gt; Hemsworth's brother, Chris, told ''[[Movieline]]'' that just a few hours after Hemsworth learned he would not appear in ''The Expendables'', director [[Kenneth Branagh]] called to ask him to test for the [[Thor (Marvel Comics)|lead role]] in the 2011 film ''[[Thor (film)|Thor]]''.&lt;ref name=&quot;chrisverge&quot;&gt;{{cite web |author=Ira Madison |url=http://www.movieline.com/2009/05/the-verge-chris-hemsworth.php |title=The Verge: Chris Hemsworth |publisher=Movieline |date=4 May 2009 |access-date=8 March 2010 |url-status=dead |archive-url=https://web.archive.org/web/20100324112935/http://www.movieline.com/2009/05/the-verge-chris-hemsworth.php |archive-date=24 March 2010}}&lt;/ref&gt; Hemsworth moved to the States in March 2009 for [[screen test]]s.&lt;ref name=&quot;vanityfair&quot;&gt;{{cite web|url=https://www.vanityfair.com/hollywood/features/2010/01/hemsworth-brothers-201001|title=The Brothers Hemsworth|last=Smith|first=Krista|date=January 2010|work=[[Vanity Fair (magazine)|Vanity Fair]]|access-date=28 February 2010|url-status=live|archive-url=https://web.archive.org/web/20100228074726/http://www.vanityfair.com/hollywood/features/2010/01/hemsworth-brothers-201001|archive-date=28 February 2010}}&lt;/ref&gt; Though he eventually lost the role to Chris in May,&lt;ref name=&quot;Nikki&quot; /&gt; Disney announced later the same week that Hemsworth had landed the part of Will Blakelee in the 2010 drama ''[[The Last Song (2010 film)|The Last Song]]'', based on the [[Nicholas Sparks (author)|Nicholas Sparks]] novel [[The Last Song (2010 film)|by the same name]]. Hemsworth plays the love interest of [[Miley Cyrus]]'s character in the film.&lt;ref&gt;{{cite web | last = Conn | first = Leslie | title = Casting for Miley Cyrus movie under way | work = [[Savannah Morning News]] | date = 19 May 2009 | url = http://www.savannahnow.com/node/724581 | access-date = 18 July 2009 | url-status=dead | archive-url = https://web.archive.org/web/20090522161449/http://savannahnow.com/node/724581 | archive-date = 22 May 2009 | df = dmy-all }}&lt;/ref&gt; [[Nikki Finke]] reported that Hemsworth had been in [[Los Angeles]] for just three weeks and had not yet found an agent when he was cast.&lt;ref name=&quot;Nikki&quot;&gt;{{cite web | last = Finke | first = Nikki | author-link = Nikki Finke | title = Exclusive: Chris Hemsworth is 'Thor' | website=[[Deadline Hollywood]] | date = 16 June 2009 | url = http://www.deadlinehollywooddaily.com/exclusive-chris-hemsworth-is-thor/ | access-date = 7 June 2009 | url-status=live | archive-url = https://web.archive.org/web/20110517074342/http://www.deadline.com/2009/05/exclusive-chris-hemsworth-is-thor/ | archive-date = 17 May 2011 | df = dmy-



```
all }}&lt;/ref&gt; Hemsworth next appeared in the music video for Cyrus' &quot;[[When I Look at You]]&quot;&lt;ref&gt;{{cite web | title = Savannah, 'The Last Song' featured in Miley Cyrus music video
1993    <title>Template:Sylvester Stallone</title>
1994 | name = Sylvester Stallone
1995 |title = Films by [[Sylvester Stallone]]
1996 * [[Sylvester Stallone filmography|Filmography]]
1997 * [[List of awards and nominations received by Sylvester Stallone|Awards and nominations]]
1998 The shop receives about 1 million visitors a year.&lt;ref name=Jamieson-1999/&gt; The shop has had many notable customers over the years. Visitors have included [[Teddy Roosevelt]], [[J. Edgar Hoover]], [[Jack Dempsey]], [[Charlie Chaplin]], [[Red Skelton]], [[John Wayne]], [[Katharine Hepburn]], [[James Van Der Beek]], and [[Sylvester Stallone]]. Cartoonist [[Robert Ripley]] of &quot;[[Ripley's Believe It or Not!]]&quot; bought totem poles and other crafts for his New York estate. The store's logs show that Queen [[Marie of Romania]] visited and &quot;sat in the Chinese chair&quot; and that [[Louis Tiffany]] bought &quot;curios, idols and a mammoth tusk,&quot;&lt;ref name=Jamieson-1999/&gt;&lt;ref name=Broom-2/&gt;
1999 |'''''[[Get Carter (2000 film)|Get Carter]]'''''||[[Stephen Kay]]||[[Sylvester Stallone]], [[Miranda Richardson]], [[Michael Caine]]||United States||Action thriller&lt;ref&gt;{{cite web|url=http://allmovie.com/work/get-carter-214069|work=Allmovie|title=Get Carter : Overview|accessdate=September 4, 2009|author=Deming, Mark}}&lt;/ref&gt;
2000 |'''''[[Driven (2001 film)|Driven]]'''''||[[Renny Harlin]]||[[Sylvester Stallone]], [[Burt Reynolds]], [[Kip Pardue]]||United States||&lt;ref&gt;{{cite web|url=http://allmovie.com/work/driven-241439|work=Allmovie|title=Driven: The Movie: Overview|accessdate=September 4, 2009|author=Williams, Karl}}&lt;/ref&gt;
2001 |'''''[[Rambo (2008 film)|Rambo]]'''''||[[Sylvester Stallone]]||[[Sylvester Stallone]], [[Julie Benz]], [[Matthew Marsden]]||United States||&lt;ref&gt;{{cite web|url=http://www.allmovie.com/work/rambo-389422|title=Rambo: Overview|date=September 8, 2009|work=Allmovie|author=Buchanan, Jason}}&lt;/ref&gt;
2002 |'''''[[Stop! Or My Mom Will Shoot]]'''''||[[Roger Spottiswoode]]||[[Sylvester Stallone]], [[Estelle Getty]]||United States||Action comedy&lt;ref&gt;{{cite web|url=http://allmovie.com/work/stop-or-my-mom-will-shoot-46981|work=Allmovie|title=Stop! or My Mom Will Shoot: Overview|access-date=September 16, 2009|author=Deming, Mark}}&lt;/ref&gt;
2003 |'''''[[Cliffhanger (film)|Cliffhanger]]'''''||[[Renny Harlin]]||[[Sylvester Stallone]], [[John Lithgow]], [[Michael Rooker]]||United States||&lt;ref&gt;{{cite web|url=http://www.allmovie.com/work/cliffhanger-9983|work=Allmovie|title=Cliffhanger: Overview|access-date=September 17, 2009|author=Brenner, Paul}}&lt;/ref&gt;
2004 | '''''[[Demolition Man (film)|Demolition Man]]''''' || [[Marco Brambilla]] || [[Sylvester Stallone]], [[Wesley Snipes]], [[Sandra Bullock]] || United States || Science fiction action&lt;ref&gt;{{cite web|url=http://www.allmovie.com/work/demolition-man-119000|work=Allmovie|title=Demolition Man: Overview|access-date=September 17, 2009|author=Brenner, Paul}}&lt;/ref&gt;
2005 |'''''[[The Specialist]]'''''||[[Luis Llosa]]||[[Sylvester Stallone]], [[Sharon Stone]], [[James Woods]]||United States||&lt;ref&gt;{{cite web|url=http://allmovie.com/work/the-specialist-133419|work=Allmovie|title=The Shadow: Overview|access-date=September 17, 2009|author=Williams, Karl}}&lt;/ref&gt;
2006 |'''''[[Assassins (1995 film)|Assassins]]'''''||[[Richard Donner]]||[[Sylvester Stallone]], [[Antonio Banderas]], [[Julianne Moore]]||United States||&lt;ref&gt;{{cite web|url=http://allmovie.com/work/assassins-135824|work=Allmovie|title=Assassins: Overview|access-date=September 18, 2009|author=Kaye, Don}}&lt;/ref&gt;
2007 |'''''[[Judge Dredd (film)|Judge Dredd]]'''''||[[Danny Cannon]]||[[Sylvester Stallone]], [[Armand Assante]], [[Diane Lane]]||United States||Science fiction action&lt;ref&gt;{{cite web|url=http://allmovie.com/work/judge-dredd-134919|work=Allmovie|title=Judge Dredd: Overview|access-date=September 18, 2009|author=Blaise, Judd}}&lt;/ref&gt;
2008 |'''''[[Daylight (1996 film)|Daylight]]'''''||[[Rob Cohen]]||[[Sylvester Stallone]], [[Amy Brenneman]], [[Viggo Mortensen]]||United States||&lt;ref&gt;{{cite web|url=http://www.allmovie.com/work/daylight-136614|work=Allmovie|title=Daylight: Overview|access-date=October 19, 2009|author=Fountain, Clarke}}&lt;/ref&gt;
2009 |'''''[[Nighthawks (1981 film)|Nighthawks]]'''''||[[Bruce Malmuth]]||[[Sylvester Stallone]], [[Billy Dee Williams]], [[Rutger Hauer]]||United States||Action thriller&lt;ref&gt;{{cite web |url=http://allmovie.com/work/nighthawks-35341 |work=Allmovie |title=Overview: Nighthawks |last=Rasmussen |first=Linda |date=October 26, 2009}}&lt;/ref&gt;
2010 |'''''[[First Blood]]'''''||[[Ted Kotcheff]]||[[Sylvester Stallone]], [[Brian Dennehy]], [[Richard Crenna]]||United States||Action thriller&lt;ref&gt;{{cite web |url=http://allmovie.com/work/first-blood-17472 |work=Allmovie |title=Overview: First Blood |last=Erickson |first=Hal |access-date=October 26, 2009}}&lt;/ref&gt;
2011 |'''''[[Rambo: First Blood Part II]]'''''||[[George Pan Cosmatos]]||[[Sylvester Stallone]], [[Richard Crenna]], [[Charles Napier (actor)|Charles Napier]]||United States||&lt;ref&gt;{{cite web |url=http://allmovie.com/work/rambo-first-blood-part-ii-40204 |work=Allmovie |title=Overview: Rambo II |access-date=October 26, 2009 |last=Beday |first=Jeremy}}&lt;/ref&gt;
2012 |'''''[[Cobra (1986 film)|Cobra]]'''''||[[George Pan Cosmatos]]||[[Sylvester Stallone]], [[Brigitte Nielsen]], [[Reni Santoni]]||United States||&lt;ref&gt;{{cite web |url=http://allmovie.com/work/cobra-10123 |work=Allmovie |title=Overview: Cobra |access-date=October 26, 2009 |last=Brenner |first=Paul}}&lt;/ref&gt;
2013 | '''''[[Rambo III]]''''' || [[Peter MacDonald (film director)|Peter MacDonald]] || [[Sylvester Stallone]], [[Richard Crenna]], [[Marc de Jonge]] || United States || Action thriller&lt;ref&gt;{{cite web |url=http://allmovie.com/work/rambo-iii-40200 |work=Allmovie |title=Overview: Rambo III |access-date=October 26, 2009 |last=Williams |first=Karl}}&lt;/ref&gt;
2014 | '''''[[Tango &amp; Cash]]''''' || [[Andrei Konchalovsky]] || [[Sylvester Stallone]], [[Kurt Russell]], [[Teri Hatcher]] || United States || &lt;ref&gt;{{cite web|url=http://allmovie.com/work/tango-amp-cash-48611|work=Allmovie|access-date=September 11, 2009|last=Firsching |first=Robert|title=Tango &amp; Cash: Overview}}&lt;/ref&gt;
2015 |'''''[[Death Race 2000]]'''''||[[Paul Bartel]]||[[David Carradine]], [[Simone Griffeth]], [[Sylvester Stallone]]||United States||Science fiction action&lt;ref&gt;{{cite web|url=http://allmovie.com/work/death-race-2000-12891|work=Allmovie|title=Overview:Death Race 2000|accessdate=November 3, 2009|author=Deming, Mark}}&lt;/ref&gt;
2016 * '''[[Rocky Balboa (film)|Rocky Balboa]]''' with introduction by [[Sylvester Stallone]].
2017 * &quot;[[Rambo (2008 film)|Rambo]]&quot; with on-screen text intro from Sylvester Stallone.
2018 Thompson has also served various roles outside of sports broadcasting. In 2014, she joined [[Mario Lopez]] and [[Tracey Edmonds]] as a co-host on the syndicated television show ''[[Extra (American TV program)|Extra]]''.&lt;ref name=Halverson2015&gt;{{cite web|last=Halverson|first=Matthew|url=https://www.seattlemet.com/news-and-city-life/2015/04/charissa-thompson-doesnt-care-what-you-think-of-her-april-2015|title=Charissa Thompson Doesn't Care What You Think of Her|work=[[Seattle Metropolitan|Seattle Met]]|date=April 24, 2015|access-date=March 8, 2020}}&lt;/ref&gt; She also appeared as the host of the &quot;Barstool Spelling Bee&quot; in May 2015.{{citation needed|date=January 2018}} In 2016, it was announced that Thompson would be a host on the [[Netflix]] [[reality show]] ''[[Ultimate Beastmaster]]''.&lt;ref&gt;{{cite web|last=Prudon|first=Laura|title=Netflix Orders Global Competition Series 'Ultimate Beastmaster' from Sylvester Stallone, Dave Broome|url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|magazine=[[Variety (magazine)|Variety]]|date=May 9, 2016|access-date=October 5, 2016}}&lt;/ref&gt; Thompson also joined [[Top Rank]]'s broadcasting team for the [[Manny Pacquiao vs. Jessie Vargas]] pay-per-view event.&lt;ref&gt;{{cite news|title=Stephen A. Smith, Brian Kenny, Charissa Thompson &amp; Tim Bradley Jr. to Call Pacquiao-Vargas Telecast|url=http://www.toprank.com/news/boxing-promotions-stephen-a-smith-brian-kenny-charissa-thompson-and-tim-bradley-jr-to-call-pacquiao-vs-vargas-world-championship-telecast|publisher=[[Top Rank]]|date=September 26, 2016|archive-url=https://web.archive.org/web/20160927123049/http://www.toprank.com/news/boxing-promotions-stephen-a-smith-brian-kenny-charissa-thompson-and-tim-bradley-jr-to-call-pacquiao-vs-vargas-world-championship-telecast|archive-date=September 27, 2016|access-date=January 13, 2018|url-status=dead}}&lt;/ref&gt;
2019 * Cosmo Carboni a con man played by Sylvester Stallone in the 1978 film ''[[Paradise Alley]]''
2020 Ventimiglia was born on July 8, 1977, in [[Anaheim, California]],&lt;ref name=&quot;biography&quot; /&gt; the youngest child of Carol and Peter Ventimiglia,&lt;ref name=&quot;Ventimiglia&quot;&gt;{{cite news|last=Keck |first=William |title=Yo! Ventimiglia gets into the tough-guy act |publisher=USA Today |date=December 18, 2006 |url=https://www.usatoday.com/life/people/2006-12-18-milo-ventimiglia_x.htm |access-date=December 21, 2006 |url-status=live |archive-url=https://web.archive.org/web/20070108011559/http://www.usatoday.com/life/people/2006-12-18-milo-ventimiglia_x.htm |archive-date=January 8, 2007 }}&lt;/ref&gt; a [[Vietnam veteran|Vietnam War veteran]].&lt;ref&gt;{{cite book|title=TV Guide|date=September 8-14, 2008|page=20}}&lt;/ref&gt; He has two sisters.&lt;ref&gt;{{cite web|work=Today|url=https://www.today.com/popculture/milo-ventimiglia-reveals-inspiration-behind-us-character-t109394|title=Try not to cry as Milo Ventimiglia reveals inspiration behind 'This is Us' character|author=Dawn, Randee|date=March 20, 2017|access-date=October 20, 2017}}&lt;/ref&gt; His father is of Sicilian descent and his mother has English and Scottish ancestry.&lt;ref name=&quot;biography&quot;&gt;{{cite web|url=https://www.biography.com/people/milo-ventimiglia-071017|work=Biography.com|publisher=[[The Biography Channel]]|title=Milo Ventimiglia|access-date=July 12, 2017|archive-url=https://web.archive.org/web/20171020191642/https://www.biography.com/people/milo-ventimiglia-071017|archive-date=October 20, 2017|url-status=dead}}&lt;/ref&gt; Ventimiglia has a self-described &quot;crooked mouth,&quot; having been born with damaged facial nerves causing the left side of his mouth to remain immobile - much like actor [[Sylvester Stallone]], with whom he worked in ''[[Rocky Balboa (film)|Rocky Balboa]]''.&lt;ref name=&quot;Ventimiglia&quot;/&gt;
2021 * [[The Expendables (2010 film)]], an action film by David Callaham and Sylvester Stallone
2022 * [[The Expendables (franchise)|''The Expendables'' (franchise)]], an American ensemble action film series co-written and developed by [[Sylvester Stallone]]
2023 The most visible film in this theme was [[Sylvester Stallone]]'s ''[[Rambo: First Blood Part II]]'' in 1985, which did the most to popularize the idea that U.S. POWs had been left behind after the war and that the government had no real interest in their rescue.&lt;ref name=&quot;daniel&quot;&gt;{{cite web | url=http://xroads.virginia.edu/~MA01/lundy/v/cinematic2.html | title=Left Behind: Cinematic Revisions of the Vietnam POW | author=Cathleen Lundy Daniel | publisher=University of Virginia | year=2001 | access-date=2008-01-25 | archive-url=https://web.archive.org/web/20080214022333/http://xroads.virginia.edu/~MA01/lundy/v/cinematic2.html | archive-date=2008-02-14 | url-status=dead }}&lt;/ref&gt; The Rambo character, who in this film may have been partly modeled after Bo Gritz,&lt;ref&gt;McConnell and Schweitzer, ''Inside Hanoi's Secret Archives'', p. 174.&lt;/ref&gt; was a Vietnam veteran commando still haunted by the multiple failures of the war. The pivotal moment of the film occurs when Rambo, realizing he was betrayed by the U.S. government and under torture from the Vietnamese and their Soviet allies, is put into radio communication with the officer who ordered the mission and tells him, &quot;Murdock. I'm coming to get ''you''!&quot;&lt;ref&gt;{{cite book | title=Action Movies: The Cinema of Striking Back | first=Harvey | last=O'Brien | publisher=Columbia University Press | location=New York | date=2012 | pages=46-49}} O'Brien's discussion includes [https://books.google.com/books?id=b60eW-PHMTQC&amp;pg=PA48 a still from the film of that scene].&lt;/ref&gt;&lt;ref&gt;{{cite book | title=Vietnam Images: War And Representation | editor-first=Jeffrey | editor-last=Walsh | editor2-first=James | editor2-last=Aulich | publisher=Macmillan | date=1989 | first=Robert J. | last=McKeever | chapter=American Myths and the Vietnam War | pages=52-53}}&lt;/ref&gt;  Rambo and the Norris films were commercially successful in both the United States and in Thailand and other parts of Southeast Asia, and did much to perpetuate the stock image of American prisoners held in bamboo cages.&lt;ref name=&quot;inside-170&quot;&gt;McConnell and Schweitzer, ''Inside Hanoi's Secret Archives'', pp. 165, 170.&lt;/ref&gt;
2024 In 1997, Myron produced Mike Figgis'''[[One Night Stand (1997 film)|One Night Stand]]'', starring [[Wesley Snipes]], which was released in November 1997; Disney's ''[[Mr. Magoo (film)|Mr. Magoo]]'', which was released Christmas of 1997, and ''[[Burn Hollywood Burn]]'', starring Whoopi Goldberg, Sylvester Stallone and Jackie Chan, which was released in January 1998. During the same year, Myron also executive produced ''[[Barney's Great Adventure]]'', the famous purple dinosaur's only feature, which PolyGram released in April 1998. Also in 1997, ''Variety'' named Myron one of Five Most Prolific Producers of the Year.  In 1998, he produced ''[[The Mod Squad (film)|The Mod Squad]]'', starring Claire Danes and Omar Epps, for MGM, which was released in March 1999. Later that year, he executive-produced ''Hefner'', an MOW for the USA Channel.
2025 A fan favorite in Pittsburgh, he earned the nickname &quot;Rambo&quot; due to his prodigious power &amp; Stallone-esque physique. He even appeared on his own &quot;Rambo&quot; poster&lt;ref&gt;{{cite
```



web | url=https://theathletic.com/1807582/2020/05/14/mike-diazs-enduring-legacy-with-the-pirates-thats-rambo-100-percent/?source=googlesearch?redirected=1 | title=Mike Diaz's enduring legacy with the Pirates: 'That's Rambo, 100 percent' }}&lt;/ref&gt; with the proceeds going to Pittsburgh's Children's Hospital. Following his major league career, he played four seasons in Japan, from 1989 until 1992, for the [[Lotte Orions]] (who in 1992 became the [[Chiba Lotte Marines]]). He was nicknamed &quot;Rambo-san&quot; there due to a perceived resemblance to [[Sylvester Stallone]].&lt;ref name=japan&gt;[http://www.japanbaseballdaily.com/foreignbattingDade-Fyhrie.html Japan Baseball Daily foreign batters page, D through F] {{webarchive|url=https://web.archive.org/web/20060517114613/http://www.japanbaseballdaily.com/foreignbattingDade-Fyhrie.html |date=2006-05-17 }}&lt;/ref&gt; In 1990, he became the first foreign player to catch a game in NPB in 12 years.&lt;ref name=japan /&gt;

2026 When casting ''Star Wars'', Lucas had a preference for unknown actors. He initially resisted casting Ford as Solo, since Ford had previously worked with him on ''[[American Graffiti]]'', and was therefore not unknown. Instead, the director asked Ford to assist with auditions by reading lines with other actors. However, Lucas was eventually won over by Ford, and cast him as Solo over many other actors who auditioned.{{Efn|Other actors who reportedly auditioned for the role of Han Solo include [[James Caan]], [[Chevy Chase]], [[Robert De Niro]], [[Richard Dreyfuss]], [[Steve Martin]], [[Bill Murray]], [[Jack Nicholson]], [[Nick Nolte]], [[Al Pacino]], [[Burt Reynolds]], [[Kurt Russell]], [[Sylvester Stallone]], [[John Travolta]], [[Christopher Walken]], and [[Perry King]], who later played Solo in the [[Star Wars (radio series)|radio series]].&lt;ref name=&quot;Dreams&quot;&gt;{{cite AV media |title=[[Empire of Dreams: The Story of the Star Wars Trilogy]] |medium=DVD | publisher=20th Century Fox Home Entertainment |year=2004}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Pollard |first=Andrew |date=September 29, 2013 |title=Movie News: James Caan Talks Turning Down Roles in Superman and Star Wars |url=https://www.starburstmagazine.com/movie-news-james-caan |access-date=March 3, 2024 |work=Starburst Magazine |archive-date=February 6, 2022 |archive-url=https://web.archive.org/web/20220206123130/https://www.starburstmagazine.com/movie-news-james-caan |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Evans |first=Bradford | date=February 17, 2011 |title=The Lost Roles of Bill Murray |url=http://splitsider.com/2011/02/the-lost-roles-of-bill-murray/ |archive-url=https://web.archive.org/web/20150520115025/http://splitsider.com/2011/02/the-lost-roles-of-bill-murray |archive-date=May 20, 2015 |access-date=May 25, 2015 |website=Splitsider}}&lt;/ref&gt;&lt;ref name=&quot;casting&quot;&gt;{{cite magazine |last=Russo |first=Tom |title=The Force Wasn't With Them |url=http://www.premiere.com/article.asp?section_id=6&amp;article_id=2164 |archive-url=https://web.archive.org/web/20060508222750/http://www.premiere.com/article.asp?section_id=6&amp;article_id=2164 |archive-date=May 8, 2006 |access-date=October 3, 2006 |magazine=[[Premiere (magazine)|Premiere]]}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine |last=Farr |first=John |date=September 19, 2014 |title=Bill Murray and the Roles That Got Away |url=https://huffingtonpost.com/john-farr/bill-murray-and-the-roles_b_5850434.html |url-status=live |archive-url=https://archive.today/20160111082612/http://www.huffingtonpost.com/john-farr/bill-murray-and-the-roles_b_5850434.html |archive-date=January 11, 2016 |access-date=May 25, 2015 |magazine=[[Huffington Post]]}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Denham |first=Jess |date=September 12, 2014 |title=Star Wars: Al Pacino turned down part of Han Solo over confusing script |url=https://www.independent.co.uk/arts-entertainment/films/news/star-wars-al-pacino-turned-down-part-of-han-solo-because-the-script-was-too-confusing-9723652.html |url-access=subscription |url-status=live |archive-url=https://ghostarchive.org/archive/20220524/https://www.independent.co.uk/arts-entertainment/films/news/star-wars-al-pacino-turned-down-part-of-han-solo-because-the-script-was-too-confusing-9723652.html |archive-date=May 24, 2022 |access-date=August 11, 2021 |work=[[The Independent]]}}&lt;/ref&gt;&lt;ref name=&quot;:4&quot;&gt;{{cite web|first=Alison|last=Nastasi|date=August 5, 2010|url=http://www.cinematical.com/2010/08/05/imagine-that-sly-stallone-auditioned-for-han-solo/|title=Imagine That: Sly Stallone Auditioned for Han Solo|website=Cinematical|access-date=August 6, 2012|archive-url=https://web.archive.org/web/20100806182036/http://www.cinematical.com/2010/08/05/imagine-that-sly-stallone-auditioned-for-han-solo/|archive-date=August 6, 2010}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Huver |first=Scott |date=May 25, 2018 |title=The Forgotten Han Solo Recalls Stepping Into Harrison Ford's Shoes |url=https://www.hollywoodreporter.com/movies/movie-news/star-wars-forgotten-han-solo-recalls-failed-audition-radio-drama-1114765/ |access-date=March 4, 2024 |website=The Hollywood Reporter |language=en-US |archive-date=March 4, 2024 |archive-url=https://web.archive.org/web/20240304045812/https://www.hollywoodreporter.com/movies/movie-news/star-wars-forgotten-han-solo-recalls-failed-audition-radio-drama-1114765/ |url-status=live }}&lt;/ref&gt;}} [[Mark Hamill]], who plays Solo's companion Luke Skywalker, admired Ford's ability to re-write his character's dialogue. He observed Ford crossing out lines in the script and replacing them with modified versions which he felt were more appropriate for Solo.{{Sfn|Rinzler|2007|p=615}}

2027 *[[Sylvester Stallone]] = ''[[Rambo (2008 film)|Rambo]]'' (2008), ''[[The Expendables (2010 film)|The Expendables]]'' (2010)

2028 Over the years, the Inn has become a magnet for Hollywood's elite, U.S. Presidents, royalty, aristocrats, captains of industry and celebrities visiting Palm Springs. They include [[Frank Sinatra]], [[Jerry Lewis]], [[Bob Hope]], [[Lucille Ball]],&lt;ref name=Meeks&gt;{{cite book |last=Meeks |first=Eric G. |title=P.S. I Love Lucy: The Story of Lucille Ball in Palm Springs |year=2011 | publisher=Horotio Limburger Oglethorpe |isbn=978-1468098549 |pages=45}}&lt;/ref&gt; [[Marlon Brando]], [[Arnold Schwarzenegger]] and [[Maria Shriver]], [[Sylvester Stallone]], [[Liza Minnelli]], [[Liberace]], [[Barry Manilow]], [[Kurt Russell]], [[Goldie Hawn]], [[Don Adams]], [[Rita Hayworth]], [[Pat Boone]], [[Debbie Reynolds]], [[Larry King]], [[John Travolta]], [[Sidney Sheldon]], [[David Hasselhoff]], President [[Gerald Ford]], and [[George Hamilton (actor)|George Hamilton]], just to name a few.

2029 In 2014, Zeljković was one of the judges in the [[Tvoje lice zvuči poznato (Croatian season 1)#Week 11|eleventh episode]] of the [[Tvoje lice zvuči poznato (Croatian season 1)|first season]] of ''[[Tvoje lice zvuči poznato (Croatian TV series)|Tvoje lice zvuči poznato]]'', the Croatian version of the show ''[[Your Face Sounds Familiar]]''. On January 27, 2017 it was revealed that Nives would compete in the [[Tvoje lice zvuči poznato (Croatian season 4)|fourth season]] of the show. She won only one of the 12 episodes, imitating [[Ylvis]] with the song &quot;[[The Fox (What Does the Fox Say?)]]&quot;, but she still got enough points to qualify to the final round achieving the second highest number of points. In the final, she imitated [[Anthony Kiedis]] of the [[Red Hot Chili Peppers]] with the smash hit song &quot;[[Dani California]]&quot;, which lead her to the victory.&lt;ref&gt;{{Cite web|url=https://www.vecernji.hr/showbiz/nives-celziju-pobjednica-showa-tvoje-lice-zvuci-poznato-1172800|title = Nives Celzijus pobjednica showa Tvoje lice zvuči poznato}}&lt;/ref&gt; She was the first female winner of the show. As a contestant, she was a public favourite, with performances imitating [[Keith Flint]] of [[The Prodigy]], [[Severina (singer)|Severina]], [[Zvonko Bogdan]], [[Sylvester Stallone]] from the 1976 film ''[[Rocky]]'', and many more.&lt;ref&gt;{{Cite web|url=https://www.story.hr/celebrity/nives-celzijus-zasluzena-je-pobjednica-4-sezone-showa-tlzp-374868|title=Nives Celzijus zaslužena je pobjednica 4. Sezone showa TLZP}}&lt;/ref&gt; She was the judge for the [[Tvoje lice zvuči poznato (Croatian season 5)|fifth season]], alongside the previous winners, as well as the [[Tvoje lice zvuči poznato (Croatian season 6)|sixth season]].

2030 Akimoto was a co-host for Team Japan in the reality competition show ''[[Ultimate Beastmaster]]'' which was released on Netflix on February 24, 2017.&lt;ref name=&quot;beast variety&quot;&gt;{{cite web|url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|title=Netflix Orders Global Competition Series 'Ultimate Beastmaster' from Sylvester Stallone, Dave Broome|last=Prudon|first=Laura|work=[[Variety (magazine)|Variety]]|date=May 9, 2016|access-date=October 5, 2016|archive-date=November 8, 2020|archive-url=https://web.archive.org/web/20201108100911/https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;beast deadline&quot;&gt;{{cite web|url=https://deadline.com/2016/05/netflix-ultimate-beastmaster-sylvester-stallone-competition-reality-series-1201751580/|title=Netflix Orders 'Ultimate Beastmaster' Competition Reality Series With Sylvester Stallone|last=Andreeva|first=Nellie|work=Deadline|date=May 9, 2016|access-date=October 5, 2016|archive-date=September 29, 2017|archive-url=https://web.archive.org/web/20170929000334/https://deadline.com/2016/05/netflix-ultimate-beastmaster-sylvester-stallone-competition-reality-series-1201751580/|url-status=live}}&lt;/ref&gt;

2031 '''''[[Rocky]]''''' is a 1976 [[film]] written by and starring [[Sylvester Stallone]] and directed by [[John G. Avildsen]]. It tells the [[rags-to-riches]] [[American Dream]] story of [[Rocky Balboa (character)|Rocky Balboa]], an uneducated but good-hearted [[debt collector]] for a [[loan shark]] in [[Philadelphia]]. Balboa is also a club fighter who gets a shot at the [[List of heavyweight boxing champions|world heavyweight championship]] when the scheduled contender breaks his hand. Also starring are [[Talia Shire]] as Adrian, [[Burt Young]] as Adrian's brother Paulie, [[Burgess Meredith]] as Rocky's trainer [[Mickey Goldmill]], and [[Carl Weathers]] as champion [[Apollo Creed]]. The film, made for only US$1.2 million and shot in just 28 days, was a [[sleeper hit]]; it made over US$117.2 million, won three [[Academy Award|Oscars]], including [[Academy Award for Best Picture|Best Picture]], to good reviews, and launched Stallone's career into the stratosphere. The film spawned [[Rocky (series)|five sequels]], ''[[Rocky II]]'', ''[[Rocky III|III]]'', ''[[Rocky IV|IV]]'', ''[[Rocky V|V]]'', and ''[[Rocky Balboa (film)|Rocky Balboa]]''.</text>

2032 ** ''[[Rhinestone (film)|Rhinestone]]'', screenplay by [[Phil Alden Robinson]] and [[Sylvester Stallone]], story by Phil Alden Robinson

2033 *'''[[6th Golden Raspberry Awards|1985]] ''[[Rambo: First Blood Part II]]'', screenplay by [[Sylvester Stallone]] and [[James Cameron]], story by [[Kevin Jarre]], based on characters created by [[David Morrell]]'''

2034 ** ''[[Rocky IV]]'', written by [[Sylvester Stallone]]

2035 ** ''[[Cobra (1986 film)|Cobra]]'', screenplay by [[Sylvester Stallone]], based on the novel ''Fair Game'' by [[Paula Gosling]]

2036 ** ''[[Rambo III]]'', written by [[Sylvester Stallone]] and [[Sheldon Lettich]], based on characters created by [[David Morrell]]

2037 ** ''[[Rocky V]]'', written by [[Sylvester Stallone]]

2038 ** ''[[Cliffhanger (film)|Cliffhanger]]'', screenplay by [[Michael France]] and [[Sylvester Stallone]], screen story by France, based on a premise by John Long

2039 ** ''[[Driven (2001 film)|Driven]]'', written by [[Sylvester Stallone]], Jan Skrentny and Neal Tabachnick

2040 **''[[Rambo: Last Blood]]'' - screenplay by Matthew Cirulnick and [[Sylvester Stallone]]; based on the [[John Rambo|character]] created by [[David Morrell]]

2041 *[[Sylvester Stallone]]

2042 In the late 1980s he created the special makeup effects on ''[[Rambo III]]''. For the scene in which Rambo heals himself by igniting gunpowder inside of a torso bullet wound, De Rossi rigged a device which caused flame to burst from both Sylvester Stallone's stomach and back at once. Stallone was so impressed by De Rossi's work that he hired him again for the make-up effects for ''[[Daylight (1996 film)|Daylight]]'',&lt;ref name=&quot;:0&quot; /&gt; which was shot in Rome and put Stallone's son [[Sage Stallone]] in contact with [[Lucio Fulci]].

2043 * When I searched for sources in a library database, all I discovered was that &quot;Scarlet Rose&quot; is the name of one of Sylvester Stallone's children. '''Delete''' unless sources are found prior to the end of this deletion discussion. [[User:Paul Erik|&lt;span style=&quot;font-family:Comic sans MS;&quot;&gt;Paul Erik&lt;/span&gt;]] &lt;small&gt;&lt;sup&gt;[[User_talk:Paul Erik|&lt;span style=&quot;color:blue;&quot;&gt;(talk)&lt;/span&gt;]][[Special:Contributions/Paul Erik|&lt;span style=&quot;color:green;&quot;&gt;(contribs)&lt;/span&gt;]]&lt;/sup&gt;&lt;/small&gt; 22:39, 10 January 2009 (UTC)

2044 | ''[[Watchmen (film)|Watchmen]]'' || [[Warner Bros. Pictures]] / [[Paramount Pictures]] / [[Legendary Entertainment|Legendary Pictures]] / [[DC Comics]] || [[Zack Snyder]] (director); [[David Hayter]], [[Alex Tse]] (screenplay); [[Malin Åkerman]], [[Billy Crudup]], [[Matthew Goode]], [[Carla Gugino]], [[Jackie Earle Haley]], [[Jeffrey Dean Morgan]], [[Patrick Wilson]], [[Matt Frewer]], [[Stephen McHattie]], [[Laura Mennell]], [[Danny Woodburn]], [[Greg Plitt]], [[James Michael Connor]], [[Robert Wisden]], [[Jerry Wasserman]], [[Frank Novak]], [[Garry Chalk]], Ron Fassler, [[Michael Kopsa]], [[Fulvio Cecere]], [[Ted Cole]], [[Mark Acheson]], [[Chris Gauthier]], [[Jay Brazeau]], [[Dan Payne]], [[Niall Matter]], Apollonia Vanova, [[Leah Gibson]], [[Carrie Genzel]], [[Greg Travis]], [[Salli Saffioti]], [[Sylvester Stallone]]

2045   <text bytes="41657" sha1="aqmrn0mwcsq0qlj2hp7qn4mgir9paqb" xml:space="preserve">{{Short description|2010 American action film by Sylvester Stallone}}

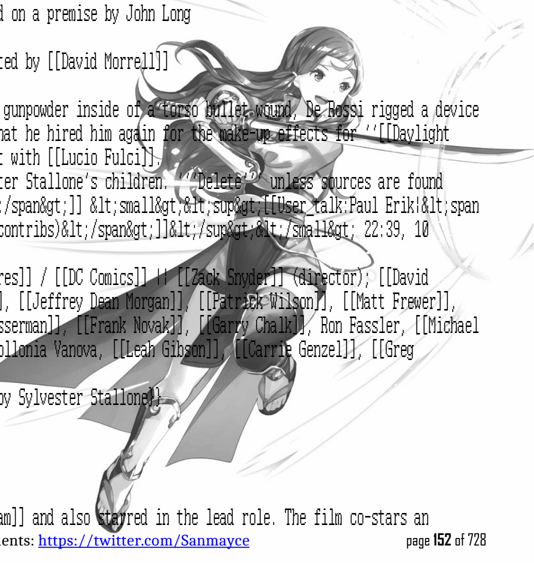
2046 | alt = Nine armed men dressed in black standing shoulder to shoulder, Sylvester Stallone front and center.

2047 | director = [[Sylvester Stallone]]

2048 * Sylvester Stallone

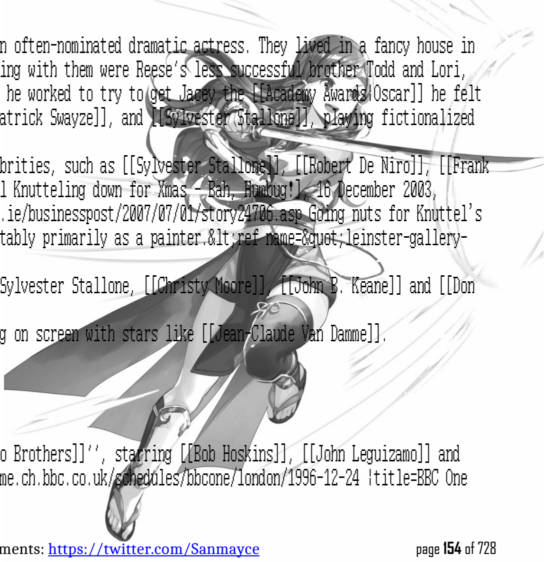
2049 * Sylvester Stallone

2050 '''''The Expendables''''' is a 2010 American [[action film]] directed by [[Sylvester Stallone]], who co-wrote it with [[David Callaham]] and also starred in the lead role. The film co-stars an



```
     [[ensemble cast]] of mostly action film actors consisting of [[Jason Statham]], [[Jet Li]], [[Dolph Lundgren]], [[Randy Couture]], [[Terry Crews]], [[Steve Austin]], [[Mickey Rourke]], and [[Bruce
     Willis]]. The film was released in the United States on [[Friday the 13th¦August 13, 2010]]. It is the first installment in [[The Expendables (film series)¦''The Expendables'' film series]]. This was
     Dolph Lundgren's first theatrically released film since 1995's cyberpunk film ''[[Johnny Mnemonic (film)¦Johnny Mnemonic]].''
2051 * [[Sylvester Stallone]] as Barney Ross
2052 In mid-2005, writer [[David Callaham]] submitted the first draft of a mercenary-inspired action film titled &quot;''Barrow''&quot; to [[Warner Bros.]], as part of his &quot;blind commitment&quot;
     deal with the studio at the time. Callaham revised the script two more times and submitted his third final revision in early 2006. Having expressed interest in doing an ensemble film, [[Sylvester
     Stallone]] reviewed Callaham's third/final revised draft of ''Barrow'' and used it as a &quot;starting point&quot; for ''The Expendables''.&lt;ref&gt;{{cite
     web¦url=http://www.loeb.com/~/media/Files/Publications/2013/01/Webb%20v%20Stallone/Files/webb%20v%20stallone/FileAttachment/webb%20v%20stallone.pdf¦title=Webb vs. Stallone¦work=United States District
     Court Southern District of New York¦access-date=October 17, 2014}}&lt;/ref&gt;
2053 The role of Hale Caesar was initially conceived as a role for Stallone's ''[[Demolition Man (film)¦Demolition Man]]'' co-star [[Wesley Snipes]]. Snipes turned down the role because of his [[Wesley
     Snipes#Income tax conviction¦tax issues]], and not able to leave the United States without the court's approval. It was later rewritten for [[Forest Whitaker]].&lt;ref
     name=&quot;SnipesCouture&quot;&gt;{{Cite news ¦ url = http://nbcsports.msnbc.com/id/28253859/ ¦ title = Couture signs on to Stallone action film ¦ date = December 16, 2008 ¦ publisher = [[NBC
     Sports]] ¦ access-date = March 20, 2009 ¦ archive-url = https://web.archive.org/web/20090402103213/http://nbcsports.msnbc.com/id/28253859/ ¦ archive-date = April 2, 2009 ¦ url-status = dead
     }}&lt;/ref&gt; Due to a scheduling conflict prior to filming, Whitaker was replaced by [[50 Cent]]&lt;ref name=&quot;WhitakerJackson&quot;&gt;{{Cite news¦ last = Sneider ¦ first = Jeff ¦ url =
     http://www.mtv.com/movies/news/articles/1607293/story.jhtml ¦ archive-url = https://web.archive.org/web/20090322080538/http://www.mtv.com/movies/news/articles/1607293/story.jhtml ¦ url-status = dead
     ¦ archive-date = March 22, 2009 ¦ title = 50 Cent To Star In Sylvester Stallone Action Film ¦ date = March 19, 2009 ¦ publisher= [[Viacom (2005-present)¦Viacom]] ¦ work = [[MTV News]] ¦ access-date =
     March 20, 2009}}&lt;/ref&gt; before the part of Hale Caesar finally went to former [[National Football League¦NFL]] player [[Terry Crews]].&lt;ref&gt;{{cite web¦ url =
     https://www.aintitcool.com/node/40516 ¦ title = 50 Cent is out! Who is in THE EXPENDABLES? Stallone says... ¦ publisher = Aintitcool.com ¦ date = 2009-03-21 ¦ access-date = 2009-04-08 }}&lt;/ref&gt;
2054 The role of the man who hires the Expendables, Mr. Church, was difficult to cast. Schwarzenegger was offered that role, but instead appeared as fellow mercenary leader Trench. The role was then
     offered to Stallone's ''[[Tango &amp; Cash]]'' co-star [[Kurt Russell]], whose agent replied that he was not interested in &quot;[[Ensemble cast¦ensemble acting]] at the moment&quot;.&lt;ref&gt;
     {{cite web ¦ url = http://www.chud.com/articles/articles/18638/1/KURT-RUSSELL-IS-NOT-EXPENDABLE/Page1.html ¦ title = Kurt Russell is Not Expendable ¦ access-date = April 14, 2009 ¦ archive-url =
     https://web.archive.org/web/20100405104838/http://www.chud.com/articles/articles/18638/1/KURT-RUSSELL-IS-NOT-EXPENDABLE/Page1.html ¦ archive-date = April 5, 2010 ¦ url-status = dead }}&lt;/ref&gt;
     Stallone spent several months after principal photography determined to find a big action name for the part. Rumors suggested that the role had been offered to friend and fellow former [[Planet
     Hollywood]] co-owner [[Bruce Willis]], who was busy filming ''[[Cop Out (2010 film)¦Cop Out]]''. Willis' casting as Mr. Church was confirmed by August 2009, as was that he would appear in a scene
     with both Stallone and Schwarzenegger.&lt;ref name=&quot;Willis&quot;&gt;{{cite web ¦ date = 2009-08-17 ¦ url= http://moviesblog.mtv.com/2009/08/17/bruce-willis-will-join-sylvester-stallone-and-
     arnold-schwarzenegger-on-screen-in-the-expendables/ ¦ archive-url= https://web.archive.org/web/20090819034223/http://moviesblog.mtv.com/2009/08/17/bruce-willis-will-join-sylvester-stallone-and-
     arnold-schwarzenegger-on-screen-in-the-expendables/ ¦ url-status= dead ¦ archive-date= August 19, 2009 ¦ title= Bruce Willis Will Join Sylvester Stallone And Arnold Schwarzenegger On Screen In 'The
     Expendables' ¦ publisher = [[Viacom (2005-present)¦Viacom]] ¦ work = Moviesblog.mtv.com ¦ access-date = 2010-03-16}}&lt;/ref&gt;
2055 To cast the soldiers, Stallone knew his filming location, Brazil, was home to many [[mixed martial arts]] fighters. He called [[Ultimate Fighting Championship]] champions [[Anderson Silva]] and
     brothers [[Antônio Rogério Nogueira¦Rogerio]] and [[Antonio Rodrigo Nogueira]] to make casting calls, and had the three cameo as well.&lt;ref&gt;[http://prommanow.com/2010/07/19/sylvester-stallone-
     talks-about-using-mma-fighters-in-the-expendables/ Sylvester Stallone talks about using MMA fighters in The Expendables]&lt;/ref&gt;
2056 [[Principal photography]] commenced 25 days later in [[Rio de Janeiro]] and other locations in [[Brazil]], including [[Mangaratiba]], [[Niteroi]], [[Guanabara Bay]], Colônia Juliano Moreira and
     [[Parque Lage]]. Filming originally ended on April 25 but was continued on May 11, in [[Elmwood, Louisiana¦Elmwood]] and [[New Orleans]], [[Louisiana]], including the [[French Quarter]], South Peters
     Street, Fort Macomb, Claiborne Avenue and the [[Interstate 10]] overpass. Filming officially ended on July 1, but on October 27 there was a [[Pick-up (filmmaking)¦pick-up]] scene at a church in [[Los
     Angeles, California]] featuring Stallone, Schwarzenegger and Willis (the latter two doing the scene without compensation, according to Stallone in the Blu-ray Disc director commentary). On June 2,
     [[West Coast Customs]] ''Street Customs'' built three customized 1955 [[Ford F-Series#Second generation (1953-1956)¦Ford F100]]s for Sylvester Stallone for the film. One was built for a crash scene,
     the second for green screen, and the third for Stallone to keep. &quot;''The Expendables'' has a seventy million dollar budget,&quot; Stallone's ''[[Rocky IV]]'' co-star [[Dolph Lundgren]] says:
     &quot;It's an old-school, kick-ass action movie where people are fighting with knives and shooting at each other.&quot;&lt;ref&gt;{{cite web¦ url= http://suicidegirls.com/interviews/Dolph+Lundgren
     %3A+Direct+Contact/ ¦ title = Dolph Lundgren: Direct Contact ¦ publisher = [[SuicideGirls.com]] ¦ date = 31 May 2009¦ access-date = 2009-06-02 }}&lt;/ref&gt; The [[flying boat]] used for filming is a
     [[Grumman HU-16 Albatross]] and the ship used as a setting in the opening scene was a Russian [[SA-15 (ship type)¦SA-15 type Arctic cargo ship]] ''Igarka''.
2057 In Summer 2010, Brazilian company [[w:pt:O2 Filmes¦O2 Filmes]] released a statement saying it was still owed more than US$2&amp;nbsp;million for its work on the film.&lt;ref&gt;{{Cite news ¦ date = 2
     August 2010 ¦ author = Tom Phillips &lt;!-- in Rio de Janeiro --&gt; ¦ title = Sylvester Stallone pursued Brazilians debts ¦ url = https://www.theguardian.com/film/2010/aug/02/sylvester-stallone-
     pursued-brazilians-debts ¦ newspaper = [[The Guardian]] ¦ location=London}}&lt;/ref&gt;
2058 The film had an original scheduled release date set at April 23, 2010, but was later pushed back four months until August 13, to extend production time.&lt;ref&gt;{{cite web ¦
     url=http://showhype.com/story/the_expendables_movies_push_back_to_august_20_2010/ ¦title=&quot;The Expendables&quot; Movies push back to August 20, 2010 - Sylvester Stallone, Celebrity Gossip ¦
     publisher=Showhype.com ¦access-date=2010-03-16}}&lt;/ref&gt; On March 17, 2010, the official international poster for the film was released.&lt;ref&gt;{{cite web ¦ url =
     https://www.denofgeek.com/movies/441783/official_poster_for_the_expendables.html ¦ title = Official Poster for The Expendables ¦ access-date = March 17, 2010 ¦ archive-url =
     https://web.archive.org/web/20120104060150/http://www.denofgeek.com/movies/441783/official_poster_for_the_expendables.html ¦ archive-date = January 4, 2012 ¦ url-status = dead }}&lt;/ref&gt; A promo
     trailer (aimed at industry professionals) was leaked online in August 2009.&lt;ref&gt;{{cite web¦last=Expendables ¦first=The ¦url=http://expendables-movie-trailer.blogspot.com/2009/09/expendables-
     trailer-leaked.html ¦title=The Expendables Trailer Leaked &amp;#124; Expendables Trailer ¦publisher=Expendables-movie-trailer.blogspot.com ¦date=2009-09-14
     ¦access-date=2010-03-16}}&lt;/ref&gt;&lt;ref&gt;{{cite web ¦ url = http://rutube.ru/tracks/2490609.html ¦ title = The Expendables Trailer ¦ access-date = November 1, 2009 ¦ archive-url =
     https://web.archive.org/web/20120528142656/http://rutube.ru/tracks/2490609.html ¦ archive-date = May 28, 2012 ¦ url-status = dead }} &lt;!-- is this really a good source? --&gt;&lt;/ref&gt; Sometime
     in October, nearly two months after the promo trailer was leaked, it was officially released online. The promo trailer was edited by Stallone and it was shown at the [[Venice Film Festival]]. On
     April 1, 2010, the official theatrical trailer for the film was released. The film had its red carpet [[Hollywood, Los Angeles¦Hollywood]] premiere on August 3, 2010. The grand premiere of the film
     was held at the [[Planet Hollywood Resort and Casino]] on the [[Las Vegas Strip]] in [[Paradise, Nevada]] on August 10, 2010.
2059 ¦ [[Sylvester Stallone]]
2060 * [[Sylvester Stallone filmography]]
2061 {{Sylvester Stallone}}
2062 [[Category:Films directed by Sylvester Stallone]]
2063 [[Category:Films with screenplays by Sylvester Stallone]]
2064 The following year, he appeared in [[James Mangold]]'s ''[[Cop Land]]'' (1997), a crime-drama co-starring [[Sylvester Stallone]], [[Harvey Keitel]] and [[Ray Liotta]]. De Niro plays [[Internal
     affairs (law enforcement)¦Internal Affairs]] investigator Lt. Moe Tilden, who explores police corruption in a New Jersey town. The film opened to a generally warm response, although Barbara
     Shulgasser of ''[[San Francisco Examiner]]'' criticized De Niro's acting in certain scenes, suggesting Mangold put De Niro in a &quot;manufactured situation&quot;, preventing him from realizing his
     full potential.&lt;ref&gt;{{Cite web¦title=A Calculated Gamble¦url=https://www.sfgate.com/news/article/A-CALCULATED-gamble-3104890.php¦last=Shulgasser¦first=Barbara¦date=August 15, 1997¦
     website=SFGate¦url-status=live¦archive-url=https://web.archive.org/web/20171207214248/http://www.sfgate.com/news/article/A-CALCULATED-gamble-3104890.php¦archive-date=December 7, 2017¦access-date=May
     10, 2020}}&lt;/ref&gt; De Niro co-starred and co-produced ''[[Wag the Dog]]'' (1997). The film is a [[political satire]] about a biased publicist (De Niro) and a Hollywood producer (Dustin Hoffman)
     who fabricate a war in Albania to cover up a U.S. president's sex scandal. In January 1998, a month after its release, the [[Clinton-Lewinsky scandal]] dominated the headlines, which helped the film
     generate publicity.&lt;ref&gt;{{Cite web¦title=Wag the Dog Back In Spotlight¦url=http://www.cnn.com/ALLPOLITICS/1998/08/21/wag.the.dog/index.html?_s=PM:ALLPOLITICS¦date=August 21, 1998¦website=CNN¦
     url-status=live¦archive-url=https://web.archive.org/web/20120915020805/http://articles.cnn.com/1998-08-21/politics/wag.the.dog_1_people-from-terrorist-activities-dogs-military-strikes?
     _s=PM:ALLPOLITICS¦archive-date=September 15, 2012¦access-date=May 11, 2020}}&lt;/ref&gt; As a result, ''Wag the Dog'' was well-received and made the list of Roger Ebert's ten best films of
     1997.&lt;ref&gt;{{Cite web¦title=The Best 10 Movies of 1997 {{!}} Roger Ebert {{!}} Roger Ebert¦url=https://www.rogerebert.com/roger-ebert/the-best-10-movies-of-1997¦last=Ebert¦first=Roger¦
     website=www.rogerebert.com¦language=en¦access-date=May 11, 2020¦archive-url=https://web.archive.org/web/20200429200539/https://www.rogerebert.com/roger-ebert/the-best-10-movies-of-1997¦archive-
     date=April 29, 2020¦url-status=live}}&lt;/ref&gt; De Niro also had a supporting role in [[Quentin Tarantino]]'s ''[[Jackie Brown]]'' in that same year.&lt;ref&gt;{{Cite news¦date=November 22, 2016¦
     title=From The Godfather to Goodfellas: Robert De Niro's greatest roles¦language=en-GB¦work=The Telegraph¦url=https://www.telegraph.co.uk/films/0/robert-de-niros-best-movies-pictures/jackie-brown/ ¦
     archive-url=https://ghostarchive.org/archive/20220110/https://www.telegraph.co.uk/films/0/robert-de-niros-best-movies-pictures/jackie-brown/ ¦archive-date=January 10, 2022 ¦url-access=subscription ¦
     url-status=live¦access-date=August 1, 2020¦issn=0307-1235}}{{cbignore}}&lt;/ref&gt;
2065 Next, he was cast in 2013's ''[[The Big Wedding]]'', [[Killing Season (film)¦''Killing Season'']], and ''[[The Family (2013 film)¦The Family]]''; all three were met with mainly a negative response.
     His other 2013 release, ''[[Last Vegas]],'' received some respectable reviews. Co-starring [[Michael Douglas]], [[Morgan Freeman]], [[Kevin Kline]] and [[Mary Steenburgen]], the film is about three
     retirees who travel to [[Las Vegas]] to have a [[bachelor party]] for their last remaining single friend. In a harsh assessment of De Niro's performance, the ''[[The A.V. Club¦A.V. Club]]''{{'s}}
     critic considered it &quot;arguably the low point of De Niro's career&quot;.&lt;ref&gt;{{Cite web¦title=Last Vegas¦url=https://film.avclub.com/last-vegas-1798178507¦last=Vishnevetsky¦first=Ignatiy¦
     date=October 31, 2013¦website=AV Club¦language=en-us¦url-status=live¦archive-url=https://web.archive.org/web/20191123151711/https://film.avclub.com/last-vegas-1798178507¦archive-date=November 23,
     2019¦access-date=May 14, 2020}}&lt;/ref&gt; Shortly afterwards, he starred in ''[[Grudge Match]]'' (2013) opposite [[Sylvester Stallone]], as aging boxers stepping into the [[Boxing ring¦ring]] for
     one last match. They had previously worked together in 1997's ''Cop Land''.&lt;ref&gt;{{Citation¦title=Grudge Match (2013)¦url=https://www.rottentomatoes.com/m/grudge_match_2014¦work=Rotten Tomatoes¦
     date=December 25, 2013 ¦language=en¦access-date=May 14, 2020¦archive-url=https://web.archive.org/web/20191106170821/https://www.rottentomatoes.com/m/grudge_match_2014/¦archive-date=November 6, 2019¦
     url-status=live}}&lt;/ref&gt; That same year he starred in the crime thriller, ''[[The Bag Man]]''. In 2014, De Niro appeared in a documentary about his father, Robert De Niro Sr., titled
     ''Remembering the Artist Robert De Niro Sr.'' which aired on [[HBO]].&lt;ref&gt;{{Cite news¦last=Shone¦first=Tom¦date=June 7, 2014¦title=Robert De Niro's thawing continues with HBO doc about late
     artist father¦work=The Guardian¦url=https://www.theguardian.com/film/2014/jun/07/remembering-artist-robert-de-niro-sr¦url-status=live¦access-date=April 17,
     2020¦archive-url=https://web.archive.org/web/20200417122742/https://www.theguardian.com/film/2014/jun/07/remembering-artist-robert-de-niro-sr¦archive-date=April 17, 2020¦issn=0261-3077}}&lt;/ref&gt;
     In 2015, he starred in [[Nancy Meyers]]' comedy ''[[The Intern (2015 film)¦The Intern]]'' alongside [[Anne Hathaway]]. The latter fared better with critics; Mark Olsen of the ''Los Angeles Times''
```

```
cordially remarked &quot;De Niro brings a fresh, relaxed lightness to his performance, tinged with the gruff charm of [[Spencer Tracy]]&quot;.&lt;ref&gt;{{Cite web|title=Review: 'The Intern' is a Nancy Meyers comedy, for better or worse|url=https://www.latimes.com/entertainment/movies/la-et-mn-the-intern-review-20150925-story.html|last=Olsen|first=Mark|date=September 25, 2015|website=Los Angeles Times|language=en-US|url-status=live|archive-url=https://web.archive.org/web/20200322105550/https://www.latimes.com/entertainment/movies/la-et-mn-the-intern-review-20150925-story.html|archive-date=March 22, 2020|access-date=May 14, 2020}}&lt;/ref&gt; His performance won him a nomination from the [[Critics Choice Movie Awards]] for [[Critics' Choice Movie Award for Best Actor in a Comedy|Best Actor in a Comedy]].&lt;ref&gt;{{cite web|url= https://www.hollywoodreporter.com/lists/2016-critics-choice-awards-winners-856447|title= Critics' Choice Awards: The Complete Winners List|website= The Hollywood Reporter|date= January 17, 2016|access-date= December 30, 2020|archive-date= April 11, 2021|archive-url= https://web.archive.org/web/20210411071053/https://www.hollywoodreporter.com/lists/2016-critics-choice-awards-winners-856447|url-status= live}}&lt;/ref&gt;
2066 |Aux1= [[Sylvester Stallone]] • [[Nicholas Hoult]] • [[Katherine Jenkins]] • [[Guillemots (band)|Guillemots]]
2067 * [[Sylvester Stallone]]&lt;ref name=&quot;Knives are out for star surgeon&quot; /&gt;
2068 In 1993, Urioste edited ''[[Cliffhanger (film)|Cliffhanger]]''.&lt;ref&gt;{{cite book|url=https://books.google.com/books?id=Y8vbAgAAQBAJ&amp;q=frank+j+urioste&amp;pg=PA269|last=Holmlund|first=Chris|title=The Ultimate Stallone Reader: Sylvester Stallone as Star, Icon, Auteur|year=2014|publisher=Columbia University Press|isbn=9780231850643}}&lt;/ref&gt;  That same year, he edited the film, ''[[Tombstone (film)|Tombstone]]''.&lt;ref&gt;{{cite book|url=https://books.google.com/books?id=RwJubSLM60EC&amp;q=frank+j+urioste&amp;pg=PT210|last2=Willis|first2=John|last1=Monush|first1=Barry|title=Screen World 1994|year=2000|publisher=Hal Leonard Corporation|isbn=9781557832023}}&lt;/ref&gt;  In 1995, he co-edited ''[[Cutthroat Island]]''.&lt;ref&gt;{{cite book|url=https://books.google.com/books?id=pO17mTyuA0IC&amp;q=frank+j+urioste&amp;pg=RA1-PA50|last=Jullius|first=Marshall|title=Action!: The Action Movie A-Z|year=1996|publisher=Indiana University Press|isbn=9780253210913}}&lt;/ref&gt;&lt;ref&gt;{{cite book|url=https://books.google.com/books?id=g6zv5YiyHtYC&amp;q=frank+j+urioste&amp;pg=PA161|last2=Willis|first2=John|last1=Monush|first1=Barry|title=Screen World 1996|year=1997|publisher=Hal Leonard Corporation|isbn=9781557832528}}&lt;/ref&gt;  In 1996, he co-edited the film, ''[[Executive Decision]]'', starring [[Kurt Russell]].&lt;ref&gt;{{cite news|last=Turan|first=Kenneth|title=Making the Right 'Decision'|date=15 March 1996|newspaper=[[Los Angeles Times]]|url=https://www.latimes.com/archives/la-xpm-1996-03-15-ca-47134-story.html|access-date=20 October 2015}}&lt;/ref&gt;
2069 ::::Since you have a computer, you're halfway there. Use it to write a half-dozen screenplays. Keep typing. Consider the early careers of [[Billy Bob Thornton]], [[Sylvester Stallone]], [[Fred Barron]], [[Harmony Korine]], [[Open Water (film)|Chris Kentis]], [[Henry Jaglom]], [[Ben Affleck]] and [[Matt Damon]]. [[User:Pepso2|Pepso2]] ([[User talk:Pepso2|talk]]) 22:44, 25 January 2009 (UTC)</text>
2070 |Co-written with [[Sylvester Stallone]]
2071 * Lieutenant Marion &quot;Cobra&quot; Cobretti - [[Paula Gosling]] (played by [[Sylvester Stallone]] in ''[[Cobra (1986 film)|Cobra]]'' )
2072 * Detective Sergeant Deke DaSilva - [[David Shaber]] and [[Paul Sylbert]] (played by [[Sylvester Stallone]] in ''[[Nighthawks (1981 film)|Nighthawks]]'' )
2073 * Lieutenant Raymond Tango - [[Randy Feldman]] and [[Jeffrey Boam]] (played by [[Sylvester Stallone]] in ''[[Tango &amp; Cash]]'' )
2074 * ''[[Oscar (1991 film)|Oscar]]'' (1991) - In a scene in the first reel, [[Sylvester Stallone]]'s daughter ([[Ornella Muti]]) is showing her rebellious side. She is in retreat in her bedroom smoking and listening to Crosby's 1932 recording of &quot;[[Sweet Georgia Brown]]&quot;.
2075 Woodley has pursued acting in his spare time and played roles in both ''[[Straight Outta Compton (film)|Straight Outta Compton]]'', ''[[Kickboxer: Vengeance]]'' and ''[[Sultan (2016 film)|Sultan]]'', the latter being an Indian movie directed by [[Ali Abbas Zafar]].&lt;ref&gt;{{Cite web|title=Tyron Woodley on Straight Outta Compton {{!}} UFC|url=https://www.ufc.com/news/tyron-woodley-straight-outta-compton|access-date=2021-06-01|website=www.ufc.com|date=September 14, 2018|language=en}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|title='Kickboxer Vengeance' Releases Teaser Trailer and Behind the Scenes Footage|url=https://www.vice.com/en/article/ezeggz/kickboxer-vengeance-releases-teaser-trailer-and-behind-the-scenes-footage|access-date=2021-06-01|website=www.vice.com|date=December 3, 2015 |language=en}}&lt;/ref&gt; He also appears in several fight scenes of ''[[Escape Plan 2: Hades]]''.&lt;ref&gt;{{Cite web|date=2017-04-12|title=Sylvester Stallone Signs On for &quot;Escape Plan 3&quot; as Titus Welliver, UFC Champ Tyron Woodley Join &quot;Part 2&quot;|url=https://www.tracking-board.com/sylvester-stallone-signs-on-for-escape-plan-3-as-titus-welliver-ufc-champ-tyron-woodley-join-part-2/|access-date=2021-06-01|website=The Tracking Board|language=en-US}}&lt;/ref&gt; Woodley also appeared on Season 5 of the [[Netflix]] show ''[[Cobra Kai]]''. On top of all of these other ventures, Woodley also appears in Season 3 Episode 5 of the [[Netflix]] show ''[[S.W.A.T._(2017_TV_series)]]'' as a member of the Jamaican Possy.
2076 |caption = North American cover art with [[Sylvester Stallone]] as [[John Rambo]]
2077 When the game was being localized for the American market, Sega bought the license to base the game on ''Rambo: First Blood Part II''. The Player 1 character was redesigned to resemble Rambo, while Player 2 was replaced with a similar character who serves as Rambo's partner. The artwork on the title screen was replaced with a reproduction of [[Sylvester Stallone]]'s pose from the film's theatrical poster, while the original title theme was replaced by a [[Programmable Sound Generator|PSG]] rendition of [[Jerry Goldsmith]]'s theme music from the movie. Some adjustments were also made to the game's difficulty (flamethrower soldiers for example, who could only could be killed using arrows in ''Ashura'', are now vulnerable to regular gunshots in ''Rambo''). The visuals on the Stage Clear screen, which originally showed various Asian landscapes and buildings in ''Ashura'', were replaced with images of Rambo in various action poses.
2078 [[Category:Images of Sylvester Stallone]]</text>
2079 Riger's television career began with Monday morning football analysis on [[Today (NBC Program)]] on NBC in 1961. In 1963, he began three years of world travel and weekly appearances on ABC's [[Wide World of Sports (U.S. TV series)]],&quot; doing the incisive on-camera picture reporting that preceded the days of color and gave ABC Sports an added dimension to their coverage. At ABC, where he worked from 1963 to 1970 and again from 1977 to 1980, he won nine Emmy awards. He developed his ground breaking slow-motion video work on the broadcasts of the 1968, 1976 and 1984 [[Summer Olympic Games]] and the 1980 [[Winter Olympic Games]]. As Second Unit Director, Riger was responsible for the acclaimed soccer sequences in the movie [[Escape to Victory]] (1981), directed by [[John Huston]], which deals with Allied prisoners of war facing German soccer stars during World War II and starring [[Sylvester Stallone]].&lt;ref&gt;{{cite web |url=http://www.hollywood.com/movie/Victory/178757 |title=Victory &amp;#124; Hollywood.com |website=www.hollywood.com |access-date=3 February 2022 |archive-url=https://archive.today/20130125113636/http://www.hollywood.com/movie/Victory/178757 |archive-date=25 January 2013 |url-status=dead}}&lt;/ref&gt;
2080 | '''[[Sylvester Stallone]]'''
2081 | [[Sylvester Stallone]]
2082 *[[Sylvester Stallone]]
2083 Fulford was featured as the murderer in the [[Cinema of the United States|Hollywood]] [[film]] ''[[D-Tox]]'' (2002, with [[Sylvester Stallone]]) and starred as the sadistic schoolmaster Metcalf in the 2002 TV adaptation of ''[[Goodbye Mr Chips]]''. He appeared in ''[[Millions (2004 film)|Millions]]'' (2004) and ''[[Pierrepoint (film)|Pierrepoint]]'' (2005) and the television dramas ''[[The Last Train (TV series)|The Last Train]]'' and ''[[Deceit (2000 miniseries)|Deceit]]''. He has also appeared extensively on [[British television]], appearing in guest roles in many episodes of series such as ''Inspector Morse'', ''[[Cracker (British TV series)|Cracker]]'', ''[[Prime Suspect]]'', ''[[Dalziel and Pascoe (TV series)|Dalziel and Pascoe]]'', ''[[Murphy's Law]]'' (fifth series), ''[[Wire in the Blood]]'' (episode &quot;Right to Silence&quot;), ''[[Spooks (TV series)|Spooks]]'', ''[[Judge John Deed]]'', ''[[Waking the Dead (TV series)|Waking the Dead]]'' and ''[[The Brief (2004 TV series)|The Brief]]''. Fulford appeared in the [[ITV1]] dramas ''[[Whitechapel (TV series)|Whitechapel]]'' and ''[[Collision (TV series)|Collision]]'', both alongside [[Phil Davis (actor)|Phil Davis]]. He played a suicidal Prime Minister in the British TV-series ''[[The Last Enemy (TV series)|The Last Enemy]]'' and in ''Crave'' for the 'Sarah Kane Season' in [[Sheffield]].
2084 [[Sylvester Stallone]]{{·}}
2085 [[Sylvester Stallone]]{{·}}
2086 [[Sylvester Stallone]]{{·}}
2087 [[Sylvester Stallone]]{{·}}
2088 [[Sylvester Stallone]]{{·}}
2089 === Sylvester Stallone to Émile Zola ===
2090 [[Sylvester Stallone]]{{·}}
2091 [[Sylvester Stallone]]{{·}}
2092 [[Sylvester Stallone]]{{·}}
2093 Reese Hardin (Hamlin) was the star of over-the-top but high-grossing action films while Jacey Watts (Grant), his second wife, was an often-nominated dramatic actress. They lived in a fancy house in [[Malibu, California]], (next door to [[Tom Hanks]], who was never seen) and raising their children, Apache and Moonglow.  Also living with them were Reese's less successful brother Todd and Lori, Reese's daughter from his first marriage. Apache always tried to come up with marketing ideas for his parents; in the second season he worked to try to get Jacey the [[Academy Awards|Oscar]] he felt she deserved. A running gag was that Todd's [[poker]] buddies were the real-life less-successful brothers of [[John Travolta]], [[Patrick Swayze]], and [[Sylvester Stallone]], playing fictionalized versions of themselves.
2094 '''Graham Knuttel''' (10 March 1954 - 27 May 2023) was an Irish painter and sculptor, whose work has been collected by various celebrities, such as [[Sylvester Stallone]], [[Robert De Niro]], [[Frank Sinatra]], [[Eddie Jordan]] and [[Michael Stipe]].&lt;ref name=&quot;circa-humbug&quot;&gt;[http://www.recirca.com/artnews/204.shtml Knutteling down for Xmas - Bah, Humbug!], 16 December 2003, retrieved 3 March 2009&lt;/ref&gt;&lt;ref name=&quot;sbp-knuttel&quot;&gt;[https://archive.today/20070812032258/http://archives.tcm.ie/businesspost/2007/07/01/story24706.asp Going nuts for Knuttel's work], [[The Sunday Business Post]], 1 July 2007, retrieved 3 March 2009&lt;/ref&gt; Initially known for sculpture, he was later notably primarily as a painter.&lt;ref name=&quot;leinster-gallery-bio&quot;/&gt;
2095 In 1981 he co-founded Wicklow Fine Art Press.&lt;ref name=&quot;irish-art-encyclopedia&quot;/&gt; He received commissions to paint Sylvester Stallone, [[Christy Moore]], [[John B. Keane]] and [[Don King (boxing promoter)|Don King]].&lt;ref name=&quot;irish-art-encyclopedia&quot;/&gt;
2096 He began doing stunts for [[Sylvester Stallone]] and then acting in movies such as ''[[Lionheart (1990 film)|Lionheart]]'', fighting on screen with stars like [[Jean-Claude Van Damme]].
2097 *[[Sylvester Stallone]], actor, screenwriter, and director
2098 {{For|the [[Sylvester Stallone]] character named Nick Martinelli|Rhinestone (film)}}
2099 [[Category:Images of Sylvester Stallone]]</text>
2100 [[Category:Images of Sylvester Stallone]]</text>
2101 *24 December - Christmas Eve highlights on BBC1 include the network television premieres of ''[[Super Mario Bros. (film)|Super Mario Brothers]]'', starring [[Bob Hoskins]], [[John Leguizamo]] and [[Dennis Hopper]] and ''[[Cliffhanger (film)|Cliffhanger]]'', starring [[Sylvester Stallone]].&lt;ref&gt;{{cite web|url=http://genome.ch.bbc.co.uk/schedules/bbcone/london/1996-12-24 |title=BBC One London - 24 December 1996 |work=BBC Genome |publisher=BBC |accessdate=2018-01-13}}&lt;/ref&gt;
2102 | colspan=2| ''[[Rambo (2008 film)|Rambo]]'' &lt;small&gt;(Director's Cut)&lt;/small&gt;|| [[Sylvester Stallone]] || United States
```



2103 * '''Jaeger-Le Coultre Glory to the Filmmaker Award''': [[Sylvester Stallone]]

2104 '''Giselle Itié Ramos''' (born October 3, 1981) is a Mexican-Brazilian actress. In 2001, she started her career as an actress in a [[Brazilian telenovela]]. In 2009, she debuted as [[protagonist]] in the telenovela ''[[Bela, a Feia]]'', the Brazilian version&lt;ref&gt;{{cite web|url=http://veja.abril.com.br/noticia/celebridades/gisele-itie-quis-ser-feia-brasil |title=Giselle Itié: 'Quis ser a feia do Brasil' |work=[[Veja (magazine)|Veja]] |publisher=[[Editora Abril]] |date=May 27, 2010 |accessdate=October 6, 2011 |url-status=dead |archiveurl=https://web.archive.org/web/20121105001913/http://veja.abril.com.br/noticia/celebridades/gisele-itie-quis-ser-feia-brasil |archivedate=November 5, 2012 }}&lt;/ref&gt; for the [[Colombia]]n ''[[Yo soy Betty, la fea]]''. Itié also took part in the film ''[[The Expendables (2010 film)|The Expendables]]'', co-written, directed by and starring [[Sylvester Stallone]].

2105 The following year, the film ''[[The Expendables (2010 film)|The Expendables]]'' premiered, in which she played Sandra, opposite [[Sylvester Stallone]].&lt;ref&gt;{{cite web|url=http://extra.globo.com/tv-e-lazer/giselle-itie-estreia-em-hollywood-em-os-mercenarios-do-stallone-366194.html|title=Giselle Itié estreia em Hollywood em 'Os mercenários', do Stallone|publisher=Extra|website=[[Globo.com]]|date=August 14, 2010|accessdate=December 18, 2011|author=Ana Carolina de Souza}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://entretenimento.r7.com/cinema/noticias/gisele-itie-fala-sobre-seu-papel-em-os-mercenarios-20100813.html|title=Giselle Itié fala sobre seu papel em Os Mercenários|date=August 13, 2010|accessdate=December 18, 2011|publisher=R7}}&lt;/ref&gt;

2106 The SFF is the first organization that brought the existence and the importance of Sicilian cinema to the world stage by considering that [[Sicily]] is second in [[Italy]] after [[Rome]] in producing movies, that some of the best Italian directors are Sicilian ([[Giuseppe Tornatore]] won an [[Academy Awards|Academy Award]] for [[Cinema Paradiso]]), that an incredible number of Hollywood stars are also Sicilian (consider [[Frank Sinatra]], [[Vincente Minnelli]], [[Martin Scorsese]], [[Frank Capra]], [[Susan Sarandon]], [[Cyndi Lauper]], [[Sylvester Stallone]], [[Joseph Barbera]]), and that an impressive number of directors from all over the world have chosen Sicily as a site for shooting their movies. Given the enormous number of Sicilians and Sicilian descendants throughout the world tired of the unjust association of their name with the Mafia and in search of a new cultural identity as an alternative to the inaccurate stereotypes, its founder Viscuso appears to have developed the right formula for involving millions of people into a converging project and a merging association. This formula includes an international background granted by the fact that the SFF is held in the United States of America rather than in Sicily, as well as the fact that other local editions of SFF will be held in other cities in the USA and in the world. At the same time the close relationship with the mother-island is established by this same &quot;Ambassador of Culture&quot; with the creation of another extremely successful event in Sicily, the FIMO ''International Festival of Organ Music in the Historical Princedom of Castelbuono'' at its first edition in 2008.

2107 **The network television premiere of ''[[Rocky IV]]'' on ITV, starring [[Sylvester Stallone]], [[Talia Shire]], [[Carl Weathers]], [[Burt Young]], [[Dolph Lundgren]] and [[Brigitte Nielsen]].

2108 *16 May - [[Arnold Schwarzenegger]], [[Bruce Willis]] and [[Sylvester Stallone]] appear as guests on the ITV chat show ''[[Aspel &amp; Company]]''.&lt;ref&gt;{{cite web|url= http://explore.bfi.org.uk/4ce2b7c995d55 |archive-url= https://web.archive.org/web/20140322001728/http://explore.bfi.org.uk/4ce2b7c995d55 |url-status= dead |archive-date= 22 March 2014 |title=Aspel &amp; Company [16/05/93] |publisher=British Film Institute |access-date=21 March 2014}}&lt;/ref&gt; The edition is later censured by the [[Independent Television Commission]] because the actors were promoting their joint business venture [[Planet Hollywood]].&lt;ref name=&quot;Inde&quot;&gt;{{cite news|author=Adrian Turpin|title=and my next guest is... |url=https://www.independent.co.uk/news/media/and-my-next-guest-istony-blair--stars-on-the-couch-468523.html |archive-url=https://ghostarchive.org/archive/20220501/https://www.independent.co.uk/news/media/and-my-next-guest-istony-blair--stars-on-the-couch-468523.html |archive-date=2022-05-01 |url-access=subscription |url-status=live |date=4 March 2006|access-date=27 December 2009 |work=[[The Independent]] |location=London}}{{cbignore}}&lt;/ref&gt;

2109 [[Sylvester Stallone]]{{·}}

2110 Later versions of the script considered [[Sylvester Stallone]], [[Joe Pesci]], [[Mark Hamill]], [[Dwayne Johnson]], [[Kevin Bacon]], [[Jean-Claude Van Damme]] or [[Matthew McConaughey]] as the celebrity.&lt;ref&gt;{{cite web |last=Woerner |first=Meredith |date=May 10, 2016 |title=McConnaughy To Bacon: All The Deleted Zombieland Cameos |url=https://io9.gizmodo.com/5374696/mcconnaughy-to-bacon-all-the-deleted-zombieland-cameos |url-status=live |archive-url=https://web.archive.org/web/20180612143948/https://io9.gizmodo.com/5374696/mcconnaughy-to-bacon-all-the-deleted-zombieland-cameos |archive-date=June 12, 2018 |access-date=June 10, 2018 |work=[[io9]] |publisher=[[Gizmodo Media Group]]}}&lt;/ref&gt;

2111 | '''BAWRA Ho Gaya Hai Kaya?''' || [[Denise Richards]]&lt;br&gt;[[Sylvester Stallone]] || ''[[Kambakkht Ishq]]''

2112 * '''[[15th Golden Raspberry Awards|1994]] - [[Sylvester Stallone]] and [[Sharon Stone]] - ''[[The Specialist]]'' (tie)'''

2113 ** [[Burt Reynolds]] and [[Sylvester Stallone]] - ''[[Driven (2001 film)|Driven]]''

2114 ** [[Sylvester Stallone]] and his [[Rage (emotion)|impotent rage]] in ''[[Rambo: Last Blood]]''

2115 * [[Sylvester Stallone]]

2116 [[Category:Images of Sylvester Stallone]]

2117 '''Julien's Auctions''' is a privately held [[auction house]] based in [[Beverly Hills, California]] founded in 2003 by Darren Julien and co-owned with Martin Nolan. They specialize in auctions of film memorabilia, music memorabilia, sports memorabilia, Fashion and street and contemporary art. Julien's Auctions has received attention for their auctions of various high-profile items. They have handled collections and estates from entertainers such as [[Marilyn Monroe]], [[John Lennon]], [[Ringo Starr]], [[Banksy]], [[Cher]], [[Bob Mackie]], [[Michael Jackson]], [[Tompkins and Bush]], [[Slash (musician)|Slash]], [[Ronnie Wood]], [[Bill Wyman]], [[Frank Zappa]], [[Greta Garbo]], [[Nirvana (band)|Nirvana]], [[Hugh Hefner]], [[Sharon Tate]], [[Pelé]], [[Elvis Presley]], [[Burt Reynolds]], [[Johnny Cash]], [[Neil Young]], [[Sylvester Stallone]], [[Frank Sinatra]], [[Whitney Houston]], [[Prince (musician)|Prince]], [[Madonna]], [[Olivia Newton-John]], [[Bette Midler]], [[Mae West]], [[Barbra Streisand]], [[Jane Fonda]], [[Aretha Franklin]], [[Janet Jackson]], [[Amy Winehouse]], [[Elizabeth Taylor]], [[Doris Day]], [[Betty White]], [[Don McLean]] and [[Fleetwood Mac]]. The auction house's first placement was in 2009 for the sale of one of Michael Jackson's bejeweled white gloves which sold for $420,000.

2118 | 2006 || ''[[Rocky Balboa (film)|Rocky Balboa]]'' || Arranger, Orchestrator, Conductor || [[Sylvester Stallone]] ||

2119 He worked on several [[Sylvester Stallone]] movies and worked with Stallone on boxing promotions in the New York City area. Giachetti died February 3, 2016, at the age of 75 in [[Lodi, Ohio]].&lt;ref name=&quot;obit&quot;&gt;{{cite web|url=https://sports.yahoo.com/blogs/boxing/richie-giachetti--who-trained-larry-holmes-and-mike-tyson--dead-at-76-002507972.html|title=Richie Giachetti, who trained Larry Holmes and Mike Tyson, dead at 75|date=4 February 2016|work=Yahoo Sports}}&lt;/ref&gt;

2120 *'''[[1984 Golden Raspberry Awards#Worst Original Song|1984]] &quot;Drinkenstein&quot; by [[Sylvester Stallone]]-''[[Rhinestone (film)|Rhinestone]]''-music and lyrics by [[Dolly Parton]]'''

2121 **&quot;Sweet Lovin' Friends&quot; by [[Dolly Parton]] and [[Sylvester Stallone]]-''[[Rhinestone (film)|Rhinestone]]''-music and lyrics by [[Dolly Parton]]

2122 ** The network television premieres of the 1981 fantasy film ''[[Clash of the Titans (1981 film)|Clash of the Titans]]'' and the 1979 [[Sylvester Stallone]] sequel ''[[Rocky II]]'' on BBC1.&lt;ref&gt;{{cite web|url=https://genome.ch.bbc.co.uk/schedules/bbcone/london/1986-01-01|title=BBC One London - 1 January 1986 - BBC Genome|website=genome.ch.bbc.co.uk|access-date=21 October 2018}}&lt;/ref&gt;

2123 * 15 September - BBC1 screen the network television premiere of the 1982 action film ''[[First Blood]]'', starring [[Sylvester Stallone]] as Vietnam War veteran [[John Rambo]].

2124 *26 December - The network television premiere of [[John G. Avildsen]]'s 1976 boxing drama film ''[[Rocky]]'' on BBC1, starring [[Sylvester Stallone]], [[Talia Shire]], [[Burt Young]], [[Burgess Meredith]] and [[Carl Weathers]].

2125 Since then his writing has attracted a diverse array of well-known actors, including [[Kevin Bacon]], [[Ben Foster (actor)|Ben Foster]], [[Michael Caine]], [[Michael Shannon]], [[Chris Rock]], [[Orlando Bloom]], [[Christopher Plummer]], [[Til Schweiger]], [[Donald Sutherland]], [[Matthew Goode]], [[Roy Scheider]], [[Djimon Hounsou]], [[Patrick Swayze]], [[Wesley Snipes]], [[Charlie Sheen]], [[Dolph Lundgren]], [[John Lithgow]], [[Tom Berenger]], [[Peter Weller]], [[Sylvester Stallone]] and [[Nick Nolte]].

2126 He has also worked with many notable directors including [[Shekhar Kapur]] (Elizabeth), [[Andrew Davis (director)|Andrew Davis]] (Holes, The Fugitive), [[Abel Ferrara]] (King of NY), [[John Hillcoat]] (Lawless), [[William Friedkin]] (The French Connection), [[Mimi Leder]] (Deep Impact), [[Renny Harlin]] (Die Hard 2, The Misfits), [[Sylvester Stallone]] (Rocky, Rambo), [[Frederic Forestier]] (Asterix...), [[Renny Harlin]], [[Wayne Cramer]] (The Cooler), and [[Brad Anderson (director)|Brad Anderson]] (The Call). His screenplays were largely used to create the first breakthrough films - for independent companies such as Elie Samaha's Franchise Pictures, and the predominant independent company in the world, Avi Lerner's [[Millennium Media|Nu Image]].

2127 [[Category:Images of Sylvester Stallone]]</text>

2128 ''I Look to You'' had first been conceived in 2007 with recording and production taking place in 2008, as said Houston during her Los Angeles album listening party in 2009.&lt;ref name=laparty/&gt; Rumor of Houston's return to music were first circulated at the start of 2007 and were then substantiated when record label boss and close mentor [[Clive Davis]] confirmed that he would personally be involved in the project whilst speaking on ''[[The Oprah Winfrey Show]]''. In February 2009, Houston appeared on stage at Davis's &quot;Pre-Grammy Gala&quot; where she performed a four-song set comprising &quot;[[I Will Always Love You]]&quot;, &quot;[[I Believe in You and Me]]&quot;, &quot;[[It's Not Right but It's Okay]]&quot;, and a rousing version of &quot;[[I'm Every Woman]]&quot;. Onlookers included her mother [[Cissy Houston]], musicians [[Paul McCartney]], [[Jamie Foxx]], and [[Barry Manilow]] as well as actors [[Antonio Banderas]] and [[Sylvester Stallone]].&lt;ref name=pregrammy&gt;{{cite web |url=http://uk.eonline.com/uberblog/b99016_whitney_houston_we_dont_have_problem.html |title=Whitney Houston we don't have a problem |work=[[E!Online|E! Online UK]] |publisher=E! Entertainment Television |date=February 8, 2009 |access-date=June 10, 2010}}&lt;/ref&gt;

2129 Bucci had a rare career as a voice dubber during the 1970s and 1980s. He dubbed [[John Travolta]] in his earlier films as well as [[Sylvester Stallone]] in ''[[The Lords of Flatbush]]''. His character dubbing roles for television include [[Potsie Weber]] in the first two seasons of ''[[Happy Days]]'' and [[Luke Duke]] in the first five seasons of ''[[The Dukes of Hazzard]]''.&lt;ref&gt;{{cite news|url=https://www.ilfattoquotidiano.it/2020/02/18/flavio-bucci-morto-a-72-anni-lattore-che-fu-don-bastiano-nel-marchese-del-grillo-e-evangelisti-nel-divo/5709495/|title= Flavio Bucci, morto a 72 anni: da &quot;La classe operaia va in Paradiso&quot; al &quot;Marchese del Grillo&quot; e al &quot;Divo&quot;. L'ultimo cameo in un film in uscita |date=18 February 2020|publisher=ilfattoquotidiano.it|language=it|access-date=19 February 2020}}&lt;/ref&gt;

2130 &quot;The Touch&quot; was released as a double A-side single with [[&quot;Weird Al&quot; Yankovic]]'s &quot;[[Dare to Be Stupid (song)|Dare to Be Stupid]]&quot;, another song from the soundtrack album of the Transformers movie. The [[power ballad]]&lt;ref&gt;{{Cite web|url=https://www.denofgeek.com/uk/go/44642|title=24 of the Biggest and Best Movie Power Ballads|website=Denofgeek.com|access-date=April 25, 2021}}{{Dead link|date=October 2022 |bot=InternetArchiveBot |fix-attempted=yes }}&lt;/ref&gt; was also released in 1987 on Stan Bush &amp; Barrage's self-titled album. The song was inspired by a line in the movie ''[[Iron Eagle]]'', and originally written for the [[Sylvester Stallone]] film ''[[Cobra (1986 film)|Cobra]].''&lt;ref&gt;{{Cite web|url=https://www.songfacts.com/facts/stan-bush/the-touch|title=The Touch by Stan Bush - Songfacts|website=Songfacts.com|access-date=April 25, 2021}}&lt;/ref&gt;&lt;ref name=vulture&gt;Riesman, Abraham (June 26, 2014). [http://www.vulture.com/2014/06/transformers-song-history-the-touch-stan-bush-mark-wahlberg-boogie-nights.html &quot;Do You Have 'The Touch'? Learn the Bizarre History of the Greatest Transformers Song Ever&quot;]. [[Vulture (website)|Vulture]].&lt;/ref&gt; Its inclusion on the ''Transformers'' soundtrack was Bush's first exposure to the franchise.{{Citation needed|date=August 2024}}

2131 In his review for ''[[Variety (magazine)|Variety]]'', Dennis Harvey said the film was &quot;passable action trash&quot; and a &quot;silly but serviceable&quot; [[actioner]], but described the performance by Mariel Hemingway as &quot;a credible [[Lesbian|Sapphic]] [[Sylvester Stallone|Stallone]]&quot;.&lt;ref name=Variety-2006&gt;{{cite magazine|last1=Harvey |first1=Dennis |title=In Her Line of Fire |url=https://variety.com/2006/film/reviews/in-her-line-of-fire-1200516755/ |magazine=[[Variety (magazine)|Variety]] |date=April 25, 2006 |access-date=24 April 2021}}&lt;/ref&gt;

2132 * [[Sylvester Stallone]] (2015)

2133 [[Category:Images of Sylvester Stallone]]</text>

2134 | {{sort|Stallone|[[Sylvester Stallone]]}}

2135 Ruginis has appeared in such films as ''[[The Devil's Advocate (1997 film)|The Devil's Advocate]]'' with [[Al Pacino]] and [[Keanu Reeves]], ''[[Cliffhanger (film)|Cliffhanger]]'' with [[Sylvester Stallone]], ''[[Broken Arrow (1996 film)|Broken Arrow]]'' with [[John Travolta]] and [[Christian Slater]], ''[[Jumpin' Jack Flash (film)|Jumpin' Jack Flash]]'' with [[Whoopi Goldberg]], ''[[The Fast and the Furious (2001 film)|The Fast and The Furious]]'', ''[[Auto Focus]]'', and many more.

2136 | {{sort|Stallone|[[Sylvester Stallone]]}}

2137 A film based on Hamed and his relationship with trainer Brendan Ingle was announced in 2023. It will be directed by Rowan Athale, with [[Sylvester Stallone]] as an executive producer. [[Mena Massoud]] and [[Paddy Considine]] will star in the film.&lt;ref&gt;{{cite web |last=Ravindran |first=Manori |date=25 January 2023 |title=Paddy Considine, Mena Massoud Headline Boxing Drama 'Giant,' Based on True Story of Naseem Hamed |url=https://variety.com/2023/film/global/paddy-considine-mena-massoud-naseem-hamed-biopic-1235502046/ |access-date=16 May 2023 |website=Variety }}&lt;/ref&gt;

2138 |ShortSummary=Kojak resists outside pressures to protect a fellow detective ([[Sylvester Stallone]]) who accidentally shot a boy. Guest Star: Claude Earl Jones.

2139 *[[Sylvester Stallone]] (born 1946), attended in 1961&lt;ref name=&quot;Alumni&quot;/&gt;

2140 [[Category:Images of Sylvester Stallone]]</text>

2141 [[Sylvester Stallone]] starred in the 1978 film ''[[F.I.S.T. (film)|F.I.S.T.]]'', a story loosely based on the [[Teamsters Union]] (a [[labor union]] which includes truck drivers) and its then president, [[Jimmy Hoffa]].

2142 # 6,363 [&lt;.01%]: [[Sylvester Stallone]]

2143 As the video begins, Jackson tries to evade overzealous fans and interviewers (including [[the Noid]] from [[Domino's Pizza]] commercials), disguising himself as a rabbit named Spike. However, as his alter ego, he goads the fans into chasing him. During the chase, he morphs into other celebrities, including [[Sylvester Stallone]], [[Tina Turner]] and [[Pee-wee Herman]]. After finally losing the mob, he removes the costume, which comes to life and challenges him to a dance-off. As the two finish dancing, a police officer (portrayed by [[Clancy Brown]]) approaches and indicates a &quot;No Dancing&quot; sign. Jackson tries to explain the situation, but Spike has vanished. The officer sarcastically indicates that he needs Jackson's &quot;autograph&quot; on a violation ticket, which Jackson grudgingly provides. The officer departs and, as Jackson prepares to do the same, a rocky crag in the distance morphs into Spike's head; Jackson and the rock formation smile at each other.

2144 | Mother of [[Sylvester Stallone]]

2145 :Is &quot;Mother of Jade Goody&quot;, &quot;Big Brother 2002 housemate&quot;, &quot;Non-celebrity&quot;, &quot;Mother of Sylvester Stallone&quot; etc an occupation? Could you suggest another alternative wording? [[User:Dalejenkins|DJ]] 10:07, 25 August 2009 (UTC)

2146 Following up ''The Truth About Cats and Dogs'', Garofalo played the lead role in ''[[The Matchmaker (1997 film)|The Matchmaker]]'', a 1997 [[romantic comedy]] film about the misadventures of a cynical American woman who reluctantly visits Ireland; it is Garofalo's first and only lead role to date. That same year, she played a supporting role as a deputy sheriff in the drama ''[[Cop Land]]'', a police gangster film starring [[Sylvester Stallone]], [[Harvey Keitel]], [[Ray Liotta]] and [[Robert De Niro]]. In 1998, she performed her first voice-acting job playing &quot;Ursula the Artist&quot; in [[The Walt Disney Company|Disney]]'s English dub of [[Studio Ghibli]]'s ''[[Kiki's Delivery Service]]'' and briefly appeared in ''[[Permanent Midnight]]''. In 1999, she starred as &quot;The Bowler&quot; in the film ''[[Mystery Men]]'', about an underdog group of super heroes.

2147 He was named &quot;the new action star of the 90s&quot; by the International Producers' Association (IPA).&lt;ref name=&quot;new&quot;&gt;{{Cite web|url=https://news.google.com/newspapers?nid=1295&amp;dat=19920625&amp;id=LsBUAAAAIBAJ&amp;pg=6732,1846413|title=Giant leap to stardom|publisher=New Sunday Times|access-date=March 22, 2020}}&lt;/ref&gt; Robert W. Welkos of ''[[Los Angeles Times]]'' declared that McColm might be &quot;Hollywood's next generation action star,&quot; following in the footsteps of Arnold Schwarzenegger and [[Sylvester Stallone]].&lt;ref&gt;{{Cite web|url=https://news.google.com/newspapers?nid=1454&amp;dat=19961130&amp;id=FWhSAAAAIBAJ&amp;pg=6818,6314399|title=MOVIE TRENDS / What's happened to all the shoot-'em-up films? Who's the next Sly or Jean-Claude?|work=[[Los Angeles Times]]|access-date=March 22, 2020}}&lt;/ref&gt; In 1997 he appeared in the lead role in the Jack Gill-directed film ''[[The Protector (1997 film)|The Protector]]''. Portraying Kenneth James Conway, an ex-commando and a private detective,&lt;ref name=&quot;tum&quot;&gt;{{Cite web|url=http://www.the-unknown-movies.com/unknownmovies/reviews/rev150.html|title=Body Armor - The Unknown Movies|publisher=The Unknown Movies|access-date=March 22, 2020}}&lt;/ref&gt; he had a chance to impress the audiences with both his martial arts skills and a perfect musculature.&lt;ref name=&quot;tum&quot;/&gt;&lt;ref name=&quot;afl&quot;/&gt;&lt;ref name=&quot;dam&quot;&gt;{{Cite web|url=https://archive.today/20200322115514/http://viendammage.blogspot.com/2014/01/crappy-action-movie-craption-body-armor.html|title=Crappy + Action Movie = Craption! Body Armor|publisher=Dammaged Goods|access-date=March 22, 2020}}&lt;/ref&gt; In a review for ''The Unknown Movies'' McColm's role was summarized positively: &quot;He looks good, shows that he knows a fair bit about martial arts, and he does seem to be putting some effort in his performance.&quot;&lt;ref name=&quot;tum&quot;/&gt; Another favorable commentary was written by ''Action-Flix'': McColm was believed to be &quot;totally charismatic in the hero role,&quot; and his physical expression was compared to that of an &quot;80s jock.&quot;&lt;ref name=&quot;afl&quot;/&gt; Likewise, the performance was praised by a cinema-related website ''Dammaged Goods'': &quot;Stunt performer turned leading man Matt McColm plays Conway with rogue charm while flexing his muscles and martial arts prowess. Ruggedly handsome, McColm doesn't take things too seriously and is a likeable enough hero.&quot;&lt;ref name=&quot;dam&quot;/&gt;

2148 '''''Zookeeper''''' is a 2011 American [[Fantasy film|fantasy]] [[comedy film]] directed by [[Frank Coraci]], with a screenplay by [[Nick Bakay]], Rock Reuben, [[Kevin James]], Jay Scherick, and David Ronn, from a story by Scherick and Ronn, and produced by [[Todd Garner]], James, [[Adam Sandler]], [[Jack Giarraputo]], and [[Walt Becker]]. The film stars James, [[Rosario Dawson]], and [[Leslie Bibb]] with the voice roles from [[Cher]], [[Nick Nolte]], Sandler, [[Sylvester Stallone]], [[Judd Apatow]], [[Jim Breuer]], [[Jon Favreau]], [[Faizon Love]], Richie Minervini, [[Maya Rudolph]], and [[Bas Rutten]]. It is about an unlucky [[zookeeper]] and the talking animals at his zoo who break their silence to help him find love.

2149 * [[Sylvester Stallone]] as Joe, an [[African lion]]

2150 '''Martin Davidson''' (born November 7, 1939) is an American film director, producer, screenwriter, television director. After attending the [[American Academy of Dramatic Arts]], he spent four (five counting tours) years as an actor in [[Off Broadway]] shows and regional theater. His directorial debut was ''[[The Lords of Flatbush]]'' starring [[Sylvester Stallone]], [[Henry Winkler]] and [[Susan Blakely]]. He won an [[CableACE Award|ACE award]] for his film ''[[Long Gone (film)|Long Gone]]''.

2151 * [[Jennifer Flavin]] (born 1968), a former model and wife of actor Sylvester Stallone

2152 * ''[[The Specialist]]'': Ray Quick ([[Sylvester Stallone]]) and Ned Trent ([[James Woods]]), [[CIA]] explosives experts, are in charge of a mission to blow up a [[South American]] drug dealer. Filming locations were actually in [[Miami]].

2153 He is now a professional actor in movies such as Roller Coaster (2015), Filling (2016) and more recently opposite [[Sylvester Stallone]] in [[Samaritan (film)|Samaritan]] (2022).

2154 Odrick is the executive producer of three short films: &quot;Roller Coaster&quot; (2015), &quot;Filling in&quot; (2016), and &quot;Jade&quot; (2017). He acted in both &quot;Filling in&quot; and &quot;Jade,&quot; playing the role of Kevin in the former, and Justin in the latter. He made an appearance in [[Sylvester Stallone]]'s thriller film, ''[[Samaritan (film)|Samaritan]],'' which was released in 2022.&lt;ref&gt;{{cite web|url=https://deadline.com/2020/02/sylvester-stallone-samaritan-jared-odrick-michael-aaron-milligan-christian-delgrosso-wallys-wonderland-film-briefs-1202862151/|title=Sylvester Stallone Thriller 'Samaritan' Adds Jared Odrick &amp; Michael Aaron Milligan; Christian DelGrosso Joins 'Wally's Wonderland' - Film Briefs|author=Amanda N'Duka |date=February 21, 2020 }}&lt;/ref&gt;

2155 *Description: A screenshot used for promotional purposes for their original film ''[[First Blood]]'' showing [[Sylvester Stallone]] as [[John Rambo]].

2156 [[Category:Images of Sylvester Stallone]]</text>

2157 |Guests=[[Sylvester Stallone]]

2158 Upon her departure from Filmways Television after five years, Jones launched her own casting agency. Her first casting job under the banner of her own company was for the 1976 [[John Avildsen]] film, ''Rocky'', which starred [[Sylvester Stallone]].&lt;ref name=variety/&gt; Jones had previously collaborated with Avildsen for his 1973 film, ''[[Save the Tiger]]'', starring [[Jack Lemmon]].&lt;ref name=hr/&gt; ''Rocky'' would win the [[Academy Award]] for best picture as well as for best director.

2159 In their own words ''&quot;The New York based philatelic agency was the first to engage in depicting high profile icons and recognized motifs on in order to assist its clients in releasing a new breed of postage stamps, which honor pop culture and sport heroes of the day including Elvis Presley, Marilyn Monroe, Pokémon, Popeye, John Lennon, the Walt Disney cartoon characters and classic motion pictures, Jackie Chan, Barbra Streisand, Bob Dylan, Major League Baseball, NFL Superbowl'' (sic)'', and the Sylvester Stallone 'Rocky' motion pictures to name just a few.&quot;''&lt;ref&gt;[https://web.archive.org/web/20090404035918/http://igpci.com/about.cfm About IGPCi] IGPC, 2008.&lt;/ref&gt;

2160 In 2009, Herzfeld directed the 90-minute documentary ''Inferno: The Making Of The Expendables'' for his friend [[Sylvester Stallone]]. The two first worked together in 1969 on a low-budget self-produced film called ''Horses'', and later again on ''[[Cobra (1986 film)|Cobra]]'', where Herzfeld plays a goon that Stallone's character sets on fire during the film's climax. Herzfeld also directed Stallone in his 2014 film, which was produced by Seraphim Films Productions, his wife, Rebekah Chaney's company. Chaney applied for the CA Tax credit in 2011 and received the funding in 2012, with only 90 days allotted for pre-production or the credit would be awarded to another production. Chaney and Herzfeld started the process together before any cast members were assigned. Chaney is the original first producer of the film ''[[Reach Me]]''.

2161 | executive_producer = [[Sylvester Stallone]]

2162 '''''Heart of a Champion: The Ray Mancini Story''''' is a 1985 American [[television film|made-for-television]] [[biographical film|biographical]] [[sports film]] starring [[Robert Blake (actor)|Robert Blake]] and [[Doug McKeon]], directed by [[Richard Michaels]] and executive-produced by [[Sylvester Stallone]]. The film originally premiered on [[CBS]] on May 1, 1985.

2163 * {{userlinks|DIREKTOR}} - Any cancellation is made by the same user. This user doesn't accept that an Italian article can use the Italian name of the family. In an historical point of view the correct name is the Italian one because these are Italian families emigrated in Dalmatia (please read the coat of arm), it's like to use [[Sylvester Stallion]] instead of [[Sylvester Stallone]]. In a technical point of view these are &quot;vandalism&quot; or at least in the it.wikipedia we (sysops) consider them like vandalisms. It seems to be like a xenofobic attempt to cancel any connections of these families with Italy or with the Italian name. Please take any action to solve the problem. [[User:Ilario|Ilario]] ([[User talk:Ilario|talk]]) 17:51, 2 October 2009 (UTC)

2164 Known for his heavily muscular physique, his biggest influences in acting include [[Action hero|action stars]] [[Arnold Schwarzenegger]], [[Bruce Willis]], and [[Sylvester Stallone]].&lt;ref name=dj/&gt;

2165 * November 29 - [[Jackie Stallone]], born Jacqueline Labofish, astrologer, dancer, wrestling promoter and mother of Sylvester Stallone (died [[2020 in the United States|2020]])

2166 [[File:Sylvester-Stallone-2014-2 (cropped).jpg|thumb|100px|[[Sylvester Stallone]]]]

2167 ** [[Sylvester Stallone]], American actor, filmmaker and screenwriter

2168 | RTitle   = [[Sylvester Stallone]], [[Robert De Niro]], [[Carey Mulligan]], [[Jonah Hill]] and [[Jake Bugg]]

2169 | [[Sylvester Stallone]]

2170 *Furthermore the BLP argument is highly controversial: '''''Nobody''''' is suggesting that Richard Gere stuck a Gerbil up his ass, '''''Everybody''''' is suggesting that there's a false rumour about it, and with 83 book references, 5 movies and tv show references, and references to it by Richard Gere and Sylvester Stallone, there is now a rediculous amount of good references about this '''''false rumour''''' and that does not a BLP violation make. [[User:Rfwoolf|Rfwoolf]] ([[User talk:Rfwoolf|talk]]) 17:16, 31 October 2009 (UTC)

2171 <text bytes="2595" sha1="76hb5tp9qhptohmfgytjff2rie936k3" xml:space="preserve">'''Scott Ross''' is a high-profile [[private investigator]] who has worked on many notable cases such as [[Robert Blake (actor)|Robert Blake]],&lt;ref&gt;{{cite web|url=https://abcnews.go.com/Entertainment/story?id=105576&amp;page=1 |title=Mystery Shrouds Shooting Death of Actor Blake's Wife - ABC News |publisher=Abcnews.go.com |date= |accessdate=2009-10-07}}&lt;/ref&gt; [[Michael Jackson]],&lt;ref name=&quot;autogenerated2005&quot;&gt;{{cite web|url=http://www.upi.com/Entertainment_News/2005/06/16/PI-discredited-mother-of-Jacksons-accuser/UPI-76731118931753/ |title=PI discredited mother of Jackson's accuser |publisher=UPI.com |date=2005-06-16 |accessdate=2009-10-07}}&lt;/ref&gt; [[Chris Brown (American singer)|Chris Brown]],&lt;ref&gt;{{cite news|url=http://latimesblogs.latimes.com/lanow/chris_brown_and_rihanna/page/2/ |title=L.A. NOW &amp;#124; Chris Brown and Rihanna &amp;#124; Los Angeles Times |publisher=Latimesblogs.latimes.com |date= 2009-03-11|accessdate=2009-10-07}}&lt;/ref&gt; [[Sylvester Stallone]],&lt;ref name=&quot;che&quot;&gt;{{cite news |last1=CHEN |first1=JOYCE |title=Sylvester Stallone hires private investigator Scott Ross to look into son Sage Moonblood Stallone's death, 'baggies' of white powder found: report |url=https://www.nydailynews.com/entertainment/gossip/sylvester-stallone-hires-private-investigator-scott-ross-son-sage-death-article-1.1116838 |access-date=12 March 2021 |work=nydailynews.com |date=18 July 2012}}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=https://latimesblogs.latimes.com/lanow/2012/07/sylvester-stallone-hires-private-investigator.html |title=Sylvester Stallone hires PI to investigate sons's death &amp;#124; L.A. Now |publisher=Latimesblogs.latimes.com |date= 2012-07-19|accessdate=2012-07-19}}&lt;/ref&gt; [[Danny Masterson]] and [[Bill Cosby]].&lt;ref&gt;{{cite news|title=FBI 'cannot confirm nor deny' if Cosby prosecutor Kevin Steele probed in federal investigation|url=http://www.yc.news/2018/01/28/fbi-cannot-confirm-deny-cosby-prosecutor-kevin-steele-probed-federal-investigation/|work=YC.NEWS|date=28 January 2018}}&lt;/ref&gt;

2172 The company has produced over thirty features and has worked alongside onscreen talent such as [[Matthew McConaughey]], [[Johnny Depp]], [[Diane Lane]], [[Anne Hathaway]], [[Julia Roberts]], [[Nicole Kidman]], [[Tom Hardy]], [[Colin Farrell]], [[Noomi Rapace]], [[Jason Statham]], [[Sylvester Stallone]], [[Harrison Ford]], and [[Gary Oldman]] in its productions.&lt;ref&gt;{{cite web|title=IM Global Profile|url=https://www.imdb.com/company/co0208137/|website=IMDB|access-date=20 November 2017}}&lt;/ref&gt;

2173 John Avildsen says [[Sylvester Stallone]] auditioned for a supporting role. He did not get the job, but starred in ''Rocky'', the director's next film.&lt;ref&gt;{{cite web|url=http://www.popentertainment.com/features_directors/avildsen.htm|website=Pop Entertainment|title=Interview with John Avildsen}}&lt;/ref&gt;

2174 The movie was based on the [[Billy the Kid]] story. [[Sylvester Stallone]] was almost cast in the lead.&lt;ref name=&quot;pink&quot;/&gt;

2175 Trautman tracks John Rambo down to a [[Buddhist]] Monastery in the jungles of [[Thailand]] and tries to convince Rambo to join him on a mission in Afghanistan to supply [[FIM-92 Stinger|Stinger missiles]] to the Afghan [[Mujahideen]] rebels, fighting the invading Soviet armies. Rambo refuses to join, because he wants to put his violent past behind him and start a new life, and Trautman understands. Trautman undertakes to deliver the shipment himself, but is captured by the Soviets in Afghanistan and Rambo decides to launch a one-man rescue mission into the Russian prison camp so he can get his only friend back home. Once again, Trautman's warnings about Rambo's brutality fall on deaf ears, and Trautman is saved by John. Trautman's role in ''Rambo III'' has been described as &quot;a well-meaning, but naive extension of the broken and mismanaged American military system&quot;.&lt;ref name=&quot;Esquire&quot;&gt;{{Cite web|url=https://www.esquire.com/entertainment/movies/a29189408/sylvester-stallone-rambo-right-wing-politics-symbol/|title=Sylvester Stallone's Rambo Has Devolved Into a Hyperbolic Symbol of Right-Wing Politics|first=Simon|last=Abrams|date=September 23, 2019|publisher=[[Esquire (magazine)|Esquire]]}}&lt;/ref&gt;

2176 The character's fate after ''Rambo III'' was never explained in the 2008 film, ''[[Rambo (2008 film)|Rambo]]'', but because Richard Crenna died before the film was made, it can be assumed that Trautman died between ''Rambo III'' and ''Rambo''. [[Sylvester Stallone]] himself explained that the character died the same day Richard Crenna died. Even though Trautman was originally supposed to be replaced by [[James Brolin]], Stallone felt it would be disrespectful to replace Crenna.&lt;ref name=&quot;Esquire&quot; /&gt; In the fourth film, Trautman only appeared in flashbacks using archive footage from the previous films. The fourth film is dedicated to Crenna's memory.

2177 '''Delete''': Butkus was Sylvester Stallone's dog and appeared in the first two films but he is not mentioned in the target article. [[User:AdamBMorgan|AdamBMorgan]] ([[User talk:AdamBMorgan|talk]]) 18:53, 1 December 2009 (UTC)

2178 '''Delete''': Butcrease was Sylvester Stallone's dog (withe his actual name or the film dog's name); he appeared in the first two films but he is not mentioned in the target article. [[User:AdamBMorgan|AdamBMorgan]] ([[User talk:AdamBMorgan|talk]]) 18:59, 1 December 2009 (UTC)

2179 * [[Sylvester Stallone]], actor&lt;ref&gt;{{cite news |first=Nathan |last=Mercer |title=Everton 1-1 Reading |date=14 January 2007 |publisher=BBC Sport |url=http://news.bbc.co.uk/sport2/hi/football/eng_prem/6236759.stm |access-date=31 January 2008}}&lt;/ref&gt;

2180 | American film by [[Radley Metzger]], based on an [[Off-Broadway|off-Broadway play]] that included [[Sylvester Stallone]]. It features a scene of unsimulated fellatio between [[Calvin Culver]] and Gerald Grant.&lt;ref name=&quot;FM-Summer1997&quot;&gt;{{citation|last=Gallagher|first=Steve|title=The Libertine|url=http://filmmakermagazine.com/archives/issues/summer1997/metzger.php|date=Summer 1997|work=[[Filmmaker (magazine)|Filmmaker Magazine]]|access-date=24 May 2015|archive-url=https://web.archive.org/web/20150922002127/http://filmmakermagazine.com/archives/issues/summer1997/metzger.php|archive-date=22 September 2015|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;PB-201406&quot;&gt;{{cite web|last=Bentley|first=Toni|author-link=Toni Bentley|title=The Legend of Henry Paris|url=http://www.playboy.com/articles/the-legend-of-henry-paris|date=June 2014|work=Playboy|access-date=26 January 2016|archive-url=https://web.archive.org/web/20160204030128/http://www.playboy.com/articles/the-legend-of-henry-paris|archive-date=4 February 2016|url-status=dead}}&lt;/ref&gt;&lt;ref name=&quot;TB-201406&quot;&gt;{{cite web|last=Bentley|first=Toni|author-link=Toni Bentley|title=The Legend of Henry Paris|url=http://www.tonibentley.com/pdfarticles/playboy/RadleyMetzger_AuteuroftheErotic_ToniBentley.pdf|date=June 2014|website=ToniBentley.com|access-date=26 January 2016|archive-url=https://web.archive.org/web/20160201001546/http://www.tonibentley.com/pdfarticles/playboy/RadleyMetzger_AuteuroftheErotic_ToniBentley.pdf|archive-date=1 February 2016|url-status=live}}&lt;/ref&gt;

2181 | [[Sylvester Stallone]]

2182 A Bell 205 is used as a [[Mountain rescue|mountain rescue helicopter]] in the 1993 film ''[[Cliffhanger (film)|Cliffhanger]]''. The aircraft is used to locate a missing jet and then employed to find stolen money. Towards the film's end the helicopter is dangling upside down against a cliff, where the hero ([[Sylvester Stallone]]) and villain ([[John Lithgow]]) brawl on the belly of the aircraft.&lt;ref&gt;{{cite web |url=http://www.rotaryaction.com/pages/cliffhang.html |title=Cliffhanger |website=Rotary Action |access-date=14 May 2014 |archive-url= https://web.archive.org/web/20150419013412/http://www.rotaryaction.com/pages/cliffhang.html |archive-date =19 April 2015}}&lt;/ref&gt;

2183 The 2010 film ''[[The Expendables (2010 film)|The Expendables]]'' also features an Albatross as the protagonists' private airplane.&lt;ref&gt;{{cite web |url=http://www.moviehabit.com/review.php?story=exp_hm10 |title=The Expendables |last=Anderson |first=Matt |date=August 2010 |website=Movie Habit |access-date=14 December 2010 |archive-url=https://web.archive.org/web/20110102010833/http://moviehabit.com/review.php?story=exp_hm10 |archive-date=2 January 2011 |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://theenterprisereport.typepad.com/news/2010/08/exclusive-new-sylvestor-stallone-action-movie-the-expendables-uses-seaplane-pilot-with-safety-questi.html |first=Eric |last=Longabardi |title=Exclusive: New Sylvester Stallone Action Movies The Expendables Used Seaplane Pilot With Safety Question |website=The Enterprise Report |date=14 August 2010}}&lt;/ref&gt;

2184 In 2019, Uwais starred in the [[Action comedy film|action comedy]] film ''[[Stuber (film)|Stuber]]'' alongside [[Kumail Nanjiani]] and [[Dave Bautista]].&lt;ref&gt;{{cite web|url=https://www.thejakartapost.com/life/2019/06/25/iko-uwais-changes-his-game-in-stuber.html|title=Iko Uwais changes his game in 'Stuber'|last=Fachriansyah|first=Rizki|work=The Jakarta Post|date=25 June 2019|access-date=9 August 2019}}&lt;/ref&gt; In the same year, it was announced that Uwais will appear in a film called ''China Town Express'' playing a man who must fight through the gangland of New York to save his family after the disappearance of his son during a gang's killing spree.&lt;ref&gt;{{Cite web|url=http://cityonfire.com/raid-star-iko-uwais-boards-the-chinatown-express/|title='The Raid' star Iko Uwais boards the 'Chinatown Express' {{!}} cityonfire.com|last=Bona|first=J. J.|language=en-US|access-date=23 July 2019}}&lt;/ref&gt; Uwais is also attached to star in ''The Bellhop'', one of the first five films in development by [[Balboa Productions]], the new production company co-founded by [[Sylvester Stallone]] in 2018.&lt;ref&gt;{{cite web|url=https://variety.com/2019/film/features/balboa-prods-sylvester-stallone-braden-aftergood-1203275756/|title=Sylvester Stallone's Production Company Wants to Be the Blumhouse of Action Films|first=Matt|last=Donnelly|publisher=Variety|date=23 July 2019|access-date=24 June 2020}}&lt;/ref&gt;

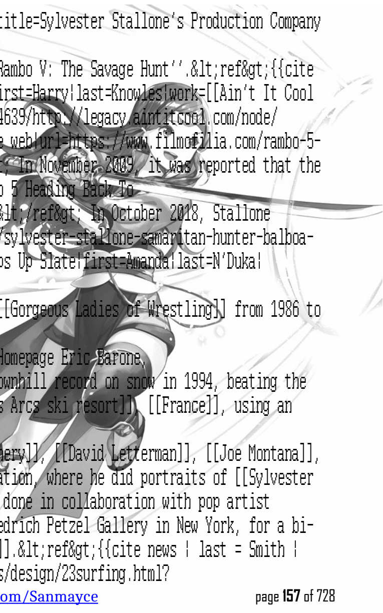
2185 |align=&quot;center&quot; |&lt;ref&gt;{{cite web|url=https://variety.com/2019/film/features/balboa-prods-sylvester-stallone-braden-aftergood-1203275756/|title=Sylvester Stallone's Production Company Wants to Be the Blumhouse of Action Films|last=Donnelly|first=Matt|work=Variety|date=23 July 2019|access-date=4 October 2019}}&lt;/ref&gt;

2186 In September 2009, [[Sylvester Stallone]] acquired the film rights to ''Hunter'' to use as the basis for ''[[Rambo: Last Blood]]'', at the time titled ''Rambo V: The Savage Hunt''.&lt;ref&gt;{{cite web|url=http://legacy.aintitcool.com/node/42256|title=UPDATED with Sly Voicemail! AICN Exclusive: RAMBO V: THE SAVAGE HUNT - a touch of science fiction|first=Harry|last=Knowles|work=[[Ain't It Cool News]]|date=September 5, 2009|access-date=August 12, 2022|url-status=live|archive-date=August 13, 2022|archive-url=https://web.archive.org/web/20220813004639/http://legacy.aintitcool.com/node/42256}}&lt;/ref&gt; Shortly after, [[Nu Image]]/Millennium Films released a concept poster and synopsis for ''Rambo V: The Savage Hunt''.&lt;ref&gt;{{cite web|url=https://www.filmofilia.com/rambo-5-the-savage-hunt-10941/|title=Rambo 5: The Savage Hunt|first=Allan|last=Ford|work=Filmofilia|date=September 8, 2009|accessdate=October 4, 2018}}&lt;/ref&gt; In November 2009, it was reported that the plot had reverted to Rambo crossing the Mexican border to rescue a girl from human traffickers.&lt;ref&gt;{{cite web|last=Outlaw|first=Kofi|title=Is Rambo 5 Heading Back To Mexico?|url=http://screenrant.com/rambo-5-script-change-mexico-plot-kofi-34228/|date=October 17, 2018|accessdate=October 24, 2018|publisher=Screen Rant}}&lt;/ref&gt; In October 2018, Stallone revealed plans to adapt ''Hunter'' into its own feature film with [[Balboa Productions]] producing.&lt;ref&gt;{{cite web|url=https://deadline.com/2018/10/sylvester-stallone-samaritan-hunter-balboa-productions-ghost-book-deal-1202483446/|title=Sylvester Stallone To Star In 'Samaritan' &amp; 'Hunter', Acquires FBI Memoir As His Balboa Productions Ramps Up Slate|first=Amanda|last=N'Duka|work=Deadline|date=October 17, 2018|accessdate=October 24, 2018}}&lt;/ref&gt;

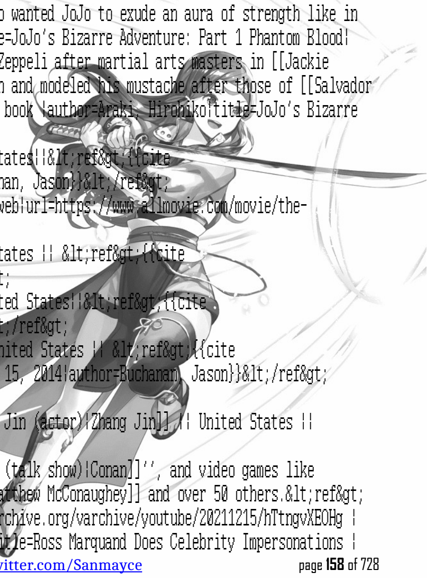
2187 | 5 || [[Jackie Stallone]] || F || {{age in years and days|1921|11|29|2020|9|21}} || November 29, 1921 || September 21, 2020 || Manager and promoter for [[Gorgeous Ladies of Wrestling]] from 1986 to 1989. Mother of [[Sylvester Stallone]].

2188 After some initial jobs, including acting as [[stunt double]] of actors [[Sylvester Stallone]], [[Jean-Claude Van Damme]] and [[Adrian Paul]],&lt;ref&gt;Homepage Eric Barone, &quot;[http://www.ericbarone.fr www.ericbarone.fr]&quot;&lt;/ref&gt; at 34 he started trying to achieve speed records on a bicycle. Barone achieved the downhill record on snow in 1994, beating the previous record by Christian Taillefer. He broke the record again in 1999 and on 21 April 2000, when he reached {{convert|222|km/h|mph}} at [[Les Arcs|Les Arcs ski resort]], [[France]], using an [[aerodynamic]] prototype bicycle, helmet and clothing.

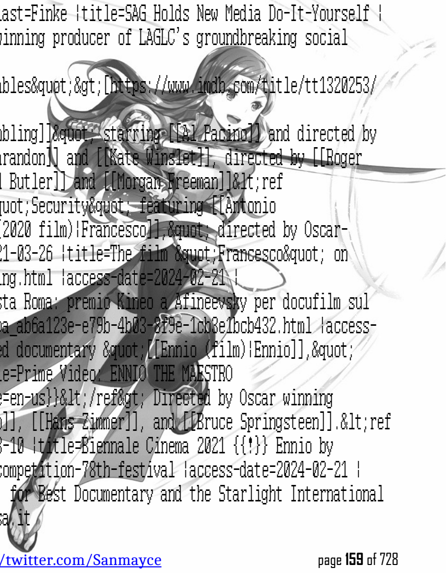
2189 An association with ''[[Playboy]]'' led to a series of silkscreen portraits of celebrities. The magazine commissioned Evans to do portraits of [[Sean Connery]], [[David Letterman]], [[Joe Montana]], and [[Marilyn Monroe]]. These portraits led to art shows in Chicago and New York, in addition to a large scale show at the Hansen Gallery Rodeo Drive location, where he did portraits of [[Sylvester Stallone]], [[Madonna]], [[Arnold Schwarzenegger]], [[Billy Idol]], [[Steve Vai]], [[Bob Dylan]], and [[Robert Mitchum]]. Later a show of feminist icons, done in collaboration with pop artist [[Richard Duardo]], had a successful run at the Zero One gallery in Los Angeles. Evans' art has been shown at The Nyehaus gallery, Metro Pictures and Friedrich Petzel Gallery in New York, for a bi-coastal tour called &quot;Swell&quot;. Other artists in the show included [[Billy Al Bengston]], [[Ed Ruscha]], [[Laddie John Dill]], and [[Dennis Hopper]].&lt;ref&gt;{{cite news | last = Smith | first = Roberta | title = Artist and Surfer as Best Buddies | work = The New York Times | date = 2010-07-22| url = https://www.nytimes.com/2010/07/23/arts/design/23surfing.html?



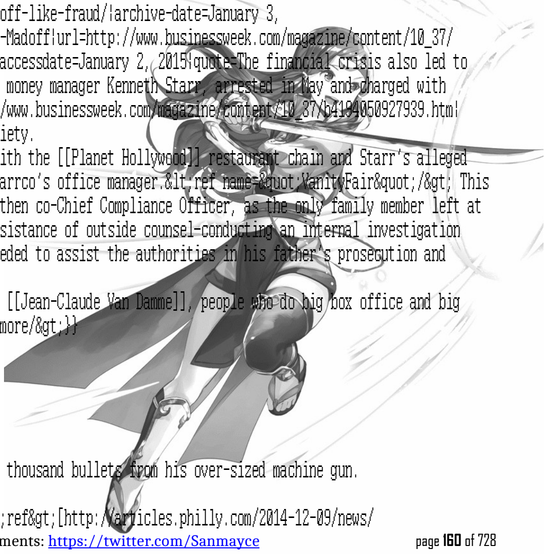
```
_r=2&amp;pagewanted=2&amp;ref=design }}&lt;/ref&gt;
2190 Lesser was born in [[New York City]], and lives in [[Santa Barbara, California|Santa Barbara]]. His earliest work dates to 1967 with &quot;[[David Holzman's Diary]]&quot;, directed by Jim McBride. He is best known for the opening scene with Bruce Willis in Die Hard, his featured role in Hester Street with Carol Kane, The Big Easy, Oscar with Sylvester Stallone, Shoot to Kill with Sidney Poitier, Presidio starring Sean Connery, the Japanese film &quot;[[Best Wishes for Tomorrow]] among others.&quot;Lesser has worked with [[Bruce Willis]], [[Ed Harris]], [[Alan Arkin]], [[Arnold Schwarzenegger]], [[Matthew Broderick]], [[Dennis Quaid]], [[Penelope Ann Miller]], and [[Richard Mulligan]].&lt;ref&gt;[https://web.archive.org/web/20121025232526/http://movies.nytimes.com/person/41986/Robert-Lesser The New York Times Movies]&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.imdb.com/name/nm0504342/|title=Robert Lesser|website=[[IMDb]]}}&lt;/ref&gt; He is well known in the Santa Barbara theater community.&lt;ref&gt;[http://www.independent.com/news/2008/apr/03/bobby-lessers-film-role-puts-america-trial Bobby Lesser's Film Role Puts America on Trial]&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://www.speakingofstories.org/Performer%20bios/robert_lesser.htm |title=ROBERT LESSER |website=www.speakingofstories.org |url-status=dead |archive-url=https://web.archive.org/web/20040912012423/http://www.speakingofstories.org/Performer%20bios/robert_lesser.htm |archive-date=2004-09-12}} &lt;/ref&gt;
2191 | ''[[The Expendables (2010 film)|The Expendables]]'' || [[Lionsgate Films|Lionsgate]] / [[Alchemy (company)|Millennium Films]] || [[Sylvester Stallone]] (director/screenplay); [[David Callaham]] (screenplay); [[Sylvester Stallone]], [[Jason Statham]], [[Jet Li]], [[Dolph Lundgren]], [[Randy Couture]], [[Steve Austin|Stone Cold Steve Austin]], [[Terry Crews]], [[Mickey Rourke]], [[Bruce Willis]], [[Eric Roberts]], [[David Zayas]], [[Giselle Itié]], [[Charisma Carpenter]], [[Gary Daniels]], [[Amin Joseph]], [[Lauren Jones]], [[Arnold Schwarzenegger]]
2192 Bloom went on to appear in the 1995 [[Sylvester Stallone]] film ''[[Daylight (1996 film)|Daylight]]'', an experience she describes as &quot;film-making at its harshest and least glamorous.&quot;
2193 In the 1990s, noted British reggae [[historiographer]], [[chronologist]] and [[archivist]] [[Steve Barrow]] and [[Simply Red]]'s [[Mick Hucknall]] collected some of Headley Bennett's [[Prince Alla]] and [[Soul Syndicate]] [[discomix]]es, drawn from ''Heaven is my Roof'' and Freedom Sounds releases, and reissued them on their [[Blood and Fire (record label)]], &lt;ref name=&quot;Meschino&quot;&gt;Meschino, Patricia (2014) &quot;[http://www.billboard.com/biz/articles/news/record-labels/5908514/vp-records-relaunches-classic-reggae-label-blood-and-fire VP Records Relaunches Classic Reggae Label Blood and Fire] {{Webarchive|url=https://web.archive.org/web/20140706021154/http://www.billboard.com/biz/articles/news/record-labels/5908514/vp-records-relaunches-classic-reggae-label-blood-and-fire |date=6 July 2014 }}&quot;, ''[[Billboard (magazine)|Billboard]]'', 19 February 2014. Retrieved 19 February 2014&lt;/ref&gt; thereby introducing Headley Bennett's work to new generations of listeners worldwide, and in 2005, aged 74, Bennett was awarded the prestigious [[Order of Distinction]] in the esteemed rank of Officer for his contribution to the development of music by the Jamaican government.&lt;ref name=&quot;NYT&quot;&gt;&quot;[https://www.nytimes.com/2005/10/18/arts/18iht-peepwed.html People: Sylvester Stallone, Headley Bennett, ABBA]&quot;, ''[[New York Times]]'', 18 October 2005, retrieved 6 December 2009&lt;/ref&gt;
2194 # 2009-12-28 10:49:58 (UTC): User [[User:80.64.84.34|80.64.84.34]] ([[User talk:80.64.84.34|talk]] - [[Special:Contributions/80.64.84.34|contribs]]; 22) to [[Sylvester Stallone]] ([http://en.wikipedia.org/w/index.php?diff=334501665&amp;oldid=334260374 diff] - [http://en.wikipedia.org/w/index.php?diff=334501665&amp;action=edit&amp;undoafter=334260374&amp;undo=334501665 undo]) - Link: neimanm.narod.ru/?page=7.&lt;!--ID 3280848--&gt;&lt;br/&gt;* Links added in this diff: &lt;nowiki&gt;neimanm.narod.ru/?page=7&lt;/nowiki&gt; (22, 22, 22, 9)
2195 # 2009-12-28 10:53:19 (UTC): User [[:w:es:User:80.64.84.34|w:es:80.64.84.34]] ([[:w:es:User talk:80.64.84.34|talk]] - [[:w:es:Special:Contributions/80.64.84.34|contribs]]; 22) to [[:w:es:Sylvester Stallone]] ([http://es.wikipedia.org/w/index.php?diff=32543685&amp;oldid=32513996 diff] - [http://es.wikipedia.org/w/index.php?diff=32543685&amp;action=edit&amp;undoafter=32513996&amp;undo=32543685 undo]) - Link: neimanm.narod.ru/?page=7.&lt;!--ID 3280994--&gt;&lt;br/&gt;* Links added in this diff: &lt;nowiki&gt;neimanm.narod.ru/?page=7&lt;/nowiki&gt; (22, 22, 22, 9)
2196 # 2009-12-28 10:56:07 (UTC): User [[:w:it:User:80.64.84.34|w:it:80.64.84.34]] ([[:w:it:User talk:80.64.84.34|talk]] - [[:w:it:Special:Contributions/80.64.84.34|contribs]]; 22) to [[:w:it:Sylvester Stallone]] ([http://it.wikipedia.org/w/index.php?diff=28899596&amp;oldid=28885392&amp;rcid=29720874 diff] - [http://it.wikipedia.org/w/index.php?diff=28899596&amp;action=edit&amp;undoafter=28885392&amp;undo=28899596 undo]) - Link: neimanm.narod.ru/?page=7.&lt;!--ID 3281104--&gt;&lt;br/&gt;* Links added in this diff: &lt;nowiki&gt;neimanm.narod.ru/?page=7&lt;/nowiki&gt; (22, 22, 22, 9)
2197 # 2009-12-28 10:58:13 (UTC): User [[:w:fr:User:80.64.84.34|w:fr:80.64.84.34]] ([[:w:fr:User talk:80.64.84.34|talk]] - [[:w:fr:Special:Contributions/80.64.84.34|contribs]]; 22) to [[:w:fr:Sylvester Stallone]] ([http://fr.wikipedia.org/w/index.php?diff=48192516&amp;oldid=48104319&amp;rcid=48764396 diff] - [http://fr.wikipedia.org/w/index.php?diff=48192516&amp;action=edit&amp;undoafter=48104319&amp;undo=48192516 undo]) - Link: neimanm.narod.ru/?page=7.&lt;!--ID 3281177--&gt;&lt;br/&gt;* Links added in this diff: &lt;nowiki&gt;neimanm.narod.ru/?page=7&lt;/nowiki&gt; (22, 22, 22, 9)
2198 *...that, inspired by [[Sylvester Stallone]]'s experience selling the script for ''[[Rocky]]'', actor/screenwriter [[J. P. Davis]] refused to sell his script for the film '''''[[Fighting Tommy Riley]]''''' unless he was guaranteed to play the lead?
2199 Sahiba started her career as a leading actress but ended in supporting roles. Her debut film was ''&quot;Mohabbat k Sodager&quot;'', released in 1992. Sahiba's first superhit film was ''Hero'' in 1992 with her future husband Jan Rambo and Izhar Qazi. Sahiba's well known films include ''Hero'' (1992), ''Hathi Mere Sathi (1993)'', ''Khazana (1995)'', ''Munda Bigra Jaye (1995)'', ''Mamla Garbar Hai (1996)'', ''Hum To Chalay Susral (1996)'', ''[[Chor Machaye Shor (1996 film)|Chor Machaye Shor]] (1996)'', ''Dil Bhi Tera Hum Bhi Tere (1997)'', ''Zever (1998)''. She has worked in both Urdu and Punjabi movies.&lt;ref&gt;http://www.citwf.com/person369936.htm, actress Sahiba on C.I.T.W.F. website, Retrieved 28 December 2015&lt;/ref&gt; She acted in many films in the 1990s but took a break after marrying fellow actor Afzal Khan, who is popularly known as Jan Rambo for his resemblance to [[John Rambo]], the character played by [[Hollywood, Los Angeles|Hollywood]] actor [[Sylvester Stallone]].&lt;ref&gt;{{cite web|url=https://tribune.com.pk/story/529144/sahiba-i-have-missed-the-big-screen|title=Sahiba: 'I have missed the big screen'|website=The Express Tribune|date=March 1, 2014}}&lt;/ref&gt; Later, Sahiba appeared in a few TV dramas with her husband.&lt;ref name=BOLNews/&gt;
2200  |Description       = Promotional image of Sylvester Stallone as Rocky Balboa
2201 Rajkumar earned the prefix 'Rambo' because of his style of realistic close combat scenes, which resembled [[Sylvester Stallone]]'s style in the movie, ''[[Rambo (film series)|Rambo]]''.&lt;ref&gt;{{cite web |url=http://www.sify.com/movies/stuntman-rambo-rajkumar-is-no-more-news-tamil-kkfsdAdbfgi.html |title=Stuntman Rambo Rajkumar is no more |website=www.sify.com |access-date=9 August 2022 |archive-url=https://web.archive.org/web/20131023174507/http://www.sify.com/movies/stuntman-rambo-rajkumar-is-no-more-news-tamil-kkfsdAdbfgi.html |archive-date=23 October 2013 |url-status=dead}}&lt;/ref&gt; Top stunt masters like [[Kanal Kannan]], Stun Siva, [[Anal Arasu]],  [[Peter Hein]], [[Stunt Silva]], [[Thalapathy Dinesh]], [[Jaguar Thangam]] &amp; [[Ram Lakshman (stunt choreographers)|Ram Laxman]] have worked as either stuntmen or stunt assistants to him.
2202 [[Sylvester Stallone]] had previously been in talks to co-star in ''Fifty/Fifty'' for [[Paramount Pictures]], first with [[Eddie Murphy]] around 1983, and later with [[Kurt Russell]].&lt;ref name=&quot;afi&quot;&gt;{{cite web | url =https://catalog.afi.com/Catalog/moviedetails/59524 | title =''Fifty/Fifty'' (1993) |author=&lt;!--Staff writer(s)/no by-line.--&gt;| website =afi.com | publisher=American Film Institute | access-date =31 December 2022}}&lt;/ref&gt; The pair eventually chose to make ''[[Tango and Cash]]'' instead. Meanwhile, writer Dennis Shryack collaborated to two Chuck Norris vehicles for producer [[Raymond Wagner]], ''[[Code of Silence (1985 film)|Code of Silence]]'' and ''[[Hero and the Terror]]'', the later made at [[Cannon Films]]. Shryack also brought with him his earlier script, ''Fifty/Fifty''.&lt;ref name=&quot;pod-trun&quot;&gt;{{cite podcast| url=https://www.podomatic.com/podcasts/explodinghelicopter/episodes/2022-02-01T04_00_00-08_00 | title=Episode 118: ''Fifty/Fifty'' | website=Exploding Helicopter | publisher=Lamb Podcasting Network | last1=Trunick | first1=Austin (guest) | date=February 1, 2022 | access-date=December 31, 2022}}&lt;/ref&gt; Cannon picked it up and went into pre-production with Chuck Norris in the Stallone role. The relaunched project was announced in the press in April 1990. Norris, however, balked at the idea of doing another jungle shoot, as his two previous works in this terrain, ''[[Missing in Action III]]'' and ''[[Delta Force 2: The Colombian Connection|Delta Force 2]]'', had been tarnished by deadly helicopter crashes.&lt;ref name=&quot;pod-trun&quot;/&gt; He was replaced by [[Peter Weller]].&lt;ref name=&quot;pod-trun&quot;/&gt; It was Robert Hays' first action film role.&lt;ref&gt;{{cite news |last=Scott |first=Vernon |date=May 5-6, 1992 |url=https://www.newspapers.com/image/613677690/ |title=Actor feels the unemployment pinch |newspaper=The Salt Lake Tribune | agency=United Press International |page=4DA {{subscription required|via=newspapers.com}}}}&lt;/ref&gt;
2203 When first beginning the series, [[Dio Brando]] is the character that Araki looked forward to drawing the most. Inspired by [[FBI method of profiling|FBI profiling]] of serial killers and how they control their victims through psychological manipulation, he gave Dio a similar trait, using his charisma to ensure his followers do his bidding. Araki revealed that he had not thought up a weakness for the character and that it was difficult to come up with a way for Dio to be defeated.&lt;ref&gt;{{cite book |author=Araki, Hirohiko|title=JoJo's Bizarre Adventure: Part 1 Phantom Blood|volume=3|page=343|date=February 3, 2015|publisher=Viz Media|isbn=978-1-4215-7881-1}}&lt;/ref&gt; As the first Joestar, Araki created [[Jonathan Joestar]] to be &quot;a symbol of purity and dignity&quot;, which he admitted might have resulted in him being &quot;boring&quot;. Jonathan's physical transformation during the seven-year skip was done with his upcoming battle with Dio in mind and inspired by muscular film actors popular at the time, such as [[Arnold Schwarzenegger]] and [[Sylvester Stallone]]. A fan of karate manga as a child, Araki also wanted JoJo to exude an aura of strength like in ''[[Karate Baka Ichidai]]'', leading to the character learning the supernatural Hamon technique.&lt;ref&gt;{{cite book |author=Araki, Hirohiko|title=JoJo's Bizarre Adventure: Part 1 Phantom Blood|volume=1|page=249|date=September 2, 2014|publisher=[[Viz Media]]|isbn=978-1-4215-7879-8}}&lt;/ref&gt; As Jonathan's teacher, Araki modeled Will A. Zeppeli after martial arts masters in [[Jackie Chan]] movies and [[Mr. Miyagi]] in ''[[The Karate Kid]]'', whose &quot;silly&quot; appearances hide their strength. Araki dressed him as a magician and modeled his mustache after those of [[Salvador Dalí]] and the character Iyami from ''[[Osomatsu-kun]]''. The Zeppeli name is derived from the English rock band [[Led Zeppelin]].&lt;ref&gt;{{cite book |author=Araki, Hirohiko|title=JoJo's Bizarre Adventure: Part 1 Phantom Blood|volume=2|page=323|date=November 4, 2014|publisher=Viz Media|isbn=978-1-4215-7880-4}}&lt;/ref&gt;
2204 |'''''[[The Expendables (2010 film)|The Expendables]]'''''||[[Sylvester Stallone]]||[[Sylvester Stallone]], [[Jason Statham]], [[Jet Li]]||United States||&lt;ref&gt;{{cite web|url=https://www.allmovie.com/work/the-expendables-474338|work=AllMovie|access-date=August 11, 2010|title=Overview: The Expendables|author=Buchanan, Jason}}&lt;/ref&gt;
2205 | '''''[[The Expendables 2]]''''' || [[Simon West]] || [[Sylvester Stallone]], [[Jason Statham]], [[Jet Li]] || United States || &lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/the-expendables-2-v537085/|work=Allmovie|title=The Expendables 2|author=Bucanan, Jason|access-date=August 18, 2012}}&lt;/ref&gt;
2206 | '''''[[Bullet to the Head]]''''' || [[Walter Hill (filmmaker)|Walter Hill]] || [[Sylvester Stallone]], [[Sung Kang]], [[Sarah Shahi]] || United States || &lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/bullet-to-the-head-v538046/|work=AllMovie|title=Bullet to the Head|access-date=February 17, 2013}}&lt;/ref&gt;
2207 | '''''[[Escape Plan (film)|Escape Plan]]''''' || [[Mikael Håfström]] || [[Sylvester Stallone]], [[Arnold Schwarzenegger]], [[James Caviezel]]||United States||&lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/escape-plan-v557537/|work=AllMovie|title=Escape Plan|access-date=October 20, 2013|author=Buchanan, Jason}}&lt;/ref&gt;
2208 | '''''[[The Expendables 3]]''''' || [[Patrick Hughes (director)|Patrick Hughes]] || [[Sylvester Stallone]], [[Mel Gibson]], [[Jason Statham]] || United States || &lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/the-expendables-3-v574178|work=AllMovie|publisher=All Media Guide|title=The Expendables 3|access-date=August 15, 2014|author=Buchanan, Jason}}&lt;/ref&gt;
2209 |[[Sylvester Stallone]], [[Dave Bautista]], [[Huang Xiaoming]], [[Jaime King]], [[50 Cent]]
2210 | '''''[[Escape Plan: The Extractors]]''''' || [[John Herzfeld]] || [[Sylvester Stallone]], [[Dave Bautista]], [[50 Cent]], [[Jaime King]], [[Zhang Jin (actor)|Zhang Jin]] || United States ||
2211 |[[Sylvester Stallone]], [[Adriana Barraza]], [[Paz Vega]], [[Yvette Monreal]], [[Sergio Peris-Mencheta]], [[Óscar Jaenada]], [[Joaquín Cosío]]
2212 An accomplished [[voiceover]] actor, Ross has lent his talents to productions such as ''[[Phineas and Ferb]]'',&lt;ref name=impress /&gt; ''[[Conan (talk show)|Conan]]'', and video games like ''[[Battlefield Hardline]]''. He is also known for impersonating a plethora of celebrities including [[Michael Caine]], [[Sylvester Stallone]], [[Matthew McConaughey]] and over 50 others.&lt;ref&gt;{{cite web |title=More Dead-On Celebrity Impressions by Ross Marquand |url=https://www.youtube.com/watch?v=hTtngvXEOHg  |archive-url=https://ghostarchive.org/varchive/youtube/20211215/hTtngvXEOHg |archive-date=December 15, 2021 |url-status=live|publisher=youtube.com |access-date=April 20, 2016}}{{cbignore}}&lt;/ref&gt;&lt;ref&gt;{{cite web |title=Ross Marquand Does Celebrity Impersonations |
```



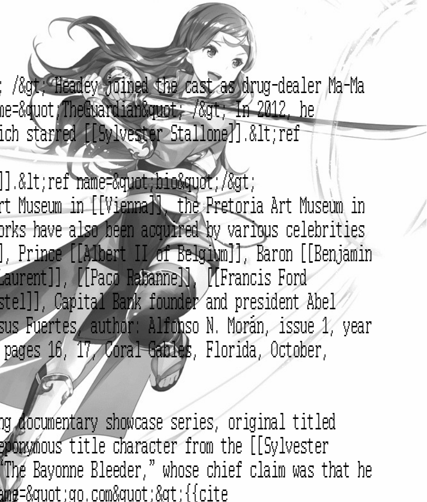
```
url=https://www.youtube.com/watch?v=8mt3M2P7qHA  |archive-url=https://ghostarchive.org/varchive/youtube/20211215/8mt3M2P7qHA |archive-date=December 15, 2021 |url-status=live|publisher=youtube.com |access-date=April 20, 2016}}{{cbignore}}&lt;/ref&gt; This expertise led to a starring and producing role on Pop TV's ''Impress Me'', helmed by Ben Shelton and [[Rainn Wilson]].&lt;ref name=impress&gt;{{cite web|first=James|last=Knight|title='Walking Dead' Gay Aaron Ross Marquand New Pop Show 'Impress Me:' Han Solo Actor Does 'Breaking Bad' Jesse Pinkman Impression|url=http://www.classicalite.com/articles/18880/20150314/walking-dead-gay-aaron-ross-marquand-new-pop-show-impress.htm|website=Classicalite|access-date=October 8, 2016|date=March 14, 2015}}&lt;/ref&gt;
2213 [[Jet Li]] is a Chinese-born Singaporean&lt;ref name=&quot;news&quot;&gt;{{cite web |title=李连杰 20 年前移民美国 被证实已经移民新加坡 |trans-title=Jet Li, emigrated to United States 20 years ago, has been confirmed migrated to Singapore |url=http://news.xinhuanet.com/audio/2009-07/29/content_11789876.htm |url-status=dead |archive-url=https://web.archive.org/web/20110609204411/http://news.xinhuanet.com/audio/2009-07/29/content_11789876.htm |archive-date=9 June 2011 |publisher=News.xinhuanet.com |access-date=2 October 2010}}&lt;/ref&gt; [[Chinese martial arts|martial artist]], retired [[wushu (sport)|wushu]] [[champion]], film actor, film producer, and philanthropist. His first non-Chinese film role was as a villain in the 1998 [[buddy cop film|buddy cop]] [[action film]] ''[[Lethal Weapon 4]]'' with [[Mel Gibson]] and [[Danny Glover]]. His first leading role in a [[Hollywood film]] was as Han Sing in the 2000 [[martial arts film|martial arts]] [[action film]] ''[[Romeo Must Die]]'' with [[Aaliyah]]. He has gone on to star in other international action films, including the [[Luc Besson]]-produced French films ''[[Kiss of the Dragon]]'' (2001) and ''[[Unleashed (2005 film)|Unleashed]]'' (2005). He co-starred in ''[[The One (2001 film)|The One]]'' (2001) and ''[[War (2007 film)|War]]'' (2007) with [[Jason Statham]], ''[[The Forbidden Kingdom]]'' (2008) with [[Jackie Chan]], the first three of ''[[The Expendables (film series)|The Expendables]]'' films with [[Sylvester Stallone]] (2010-2014), and as the title character villain in ''[[The Mummy: Tomb of the Dragon Emperor]]'' (2008). In 2020, he portrayed [[List of Disney's Mulan characters#The Emperor of China|The Emperor of China]] in the live-action [[Fantasy film|fantasy]] [[Drama (film and television)|drama]] [[Walt Disney Pictures|Disney]] film ''[[Mulan (2020 film)|Mulan]]''.
2214 |Guests=[[Sylvester Stallone]], [[Criss Angel]]
2215 |Guests=[[Sylvester Stallone]], [[Steve Wiebe]]
2216 |ShortSummary=Remote segment of [[Geoff Peterson]], &quot;live&quot;, voiced ad-lib by [[Josh Robert Thompson]] at the premiere of the movie ''[[Red (2010 film)|Red]]''. There he chats briefly with [[Morgan Freeman]],  [[Richard Dreyfus]], [[John Malkovich]], [[Mary Louise Parker]], ''Red'' director [[Taylor Hackford]], [[Sylvester Stallone]] and [[Bruce Willis]]
2217 |style=&quot;text-align:center;&quot;|[[St. Louis Gateway Film Critics Association Awards 2015|2015]] || '''[[Sylvester Stallone]]'''|| ''[[Creed (film)|Creed]]'' || [[Rocky Balboa]]
2218 Artists [[Andy Warhol]] and [[Jamie Wyeth]] had a joint exhibition at the hotel mounted by New York's Coe Kerr Gallery in July 1980.&lt;ref&gt;{{Cite web |title=Legendary Artists Andy Warhol &amp; Jamie Wyeth Sign Invitation 1980 Joint Exhibit |url=https://www.universityarchives.com/auction-lot/legendary-artists-andy-warhol-jamie-wyeth-sign_E41480FB34 |website=University Archives}}&lt;/ref&gt;&lt;ref&gt;{{Cite news |last=Smith |first=Liz |date=1980-08-18 |title=Monte Carlo's more than meets the eye |url=https://www.newspapers.com/article/daily-news-andy-warhol-jamie-wyeth-joi/149010301/ |access-date=2024-06-09 |work=Daily News |pages=7C}}&lt;/ref&gt; While Warhol was staying at the hotel, he took [[Instant film|Polaroid photos]] of actor [[Sylvester Stallone]] for his [[Screen printing|silkscreen]] portraits.&lt;ref&gt;{{Cite book |last=Warhol |first=Andy |url= |title=The Andy Warhol Diaries |last2=Hackett |first2=Pat |date=1989 |publisher=Warner Books |others= |isbn=978-0-446-51426-2 |location=New York, NY |pages=302}}&lt;/ref&gt;
2219 | rowspan=&quot;2&quot; | Website || Facebook || &quot;Ready to Rock&quot; || A series of [[List of Facebook features#Groups|Facebook groups]] containing the word &quot;rock&quot; are showcased, including cameos by [[Chris Rock]] and [[Sylvester Stallone]].&lt;ref&gt;{{Cite web|url=https://www.adweek.com/digital/sylvester-stallone-and-chris-rock-will-tout-facebook-groups-during-the-super-bowl/|title=Facebook's First Foray Into the Big Game Pushes Groups|website=Adweek|date=January 27, 2020 |language=en-US|access-date=February 9, 2020|archive-date=February 2, 2020|archive-url=https://web.archive.org/web/20200202035128/https://www.adweek.com/digital/sylvester-stallone-and-chris-rock-will-tout-facebook-groups-during-the-super-bowl/|url-status=live}}&lt;/ref&gt;
2220 | [[Paramount+]] || &quot;Stallone Face&quot;|| Characters from [[Paramount Global|Paramount's]] television and film characters assemble together to talk about climbing the Paramount Mountain, which contains a giant carving of [[Sylvester Stallone]]'s face. The real Stallone, who is hanging off the carving, falls after the carving of Stallone's face sneezes.&lt;ref&gt;{{cite web | url=https://www.usatoday.com/story/entertainment/tv/2023/02/02/sylvester-stallone-climbs-stone-head-paramount-super-bowl/11163537002/ | title=Sylvester Stallone talks new family reality show, climbing 'Paramount Mountain' in Super Bowl spot | website=[[USA Today]] }}&lt;/ref&gt;&lt;!-- 1st 2:24  --&gt;
2221 | [[Sylvester Stallone]] || ''[[Creed (film)|Creed]]'' || [[Rocky Balboa]]
2222 '''''[[First Blood]]''''' is a 1982 film starring Sylvester Stallone also known as &quot;Rambo 1&quot;.
2223 * ''[[First Blood Part II]]'', a 1985 Sylvester Stallone film also called &quot;Rambo 2&quot;
2224 |[[Paul Scheer]]&lt;small&gt; as himself and [[Frank Stallone]]&lt;/small&gt;&lt;br&gt;[[Nick Kroll]]&lt;small&gt; as himself and [[Sylvester Stallone]]&lt;/small&gt;
2225 |11. Frank and Sylvester Stallone (Episode 241: The Stallone Bros.)&lt;br&gt;BONUS: Breaking Bad Finale Spoilers (Episode 249: Ice Cold STaB)&lt;br&gt;10. The Return of the Tom Leykis Show (Episode 243: Blow Me Up Tom!)&lt;br&gt;9. Reardon and Wompler and The Bachelor Brothers (Episode 218: The 4th Anniversary Extravaganza!)&lt;br&gt;BONUS: Tito Ben's Karaoke (Episode 237: Filipino Blockbuster)&lt;br&gt;8. Werner Herzog in Jack Reacher (Episode 229: Two Thumbs &amp; Not Much Else)&lt;br&gt;BONUS: Silent Night by Bjork (Episode 191: 2012 Holiday Spectacular)&lt;br&gt;7. Richard Harrow (Episode 236: Murderer Heaven)
2226 |[[Nick Kroll]] &lt;small&gt;as himself, Liz, C-Czar, Sylvester Stallone, and El Chupacabra&lt;/small&gt; &lt;br&gt;[[Jenny Slate]] &lt;small&gt;as herself and Pretty Liz&lt;/small&gt;
2227 |[[Lauren Lapkus]] &lt;small&gt;as Todd&lt;/small&gt;&lt;br&gt;[[Colin Hanks]]&lt;br&gt;[[Nick Kroll]]&lt;small&gt; as himself, El Chupacabra, and [[Sylvester Stallone]]&lt;/small&gt;&lt;br&gt;[[Horatio Sanz]] &lt;small&gt;as [[Kanye West]]&lt;/small&gt;
2228 |[[Ben Schwartz]]&lt;br&gt;[[Jon Gabrus]] &lt;small&gt;as Gino Lombardo&lt;/small&gt;&lt;br&gt;[[Nick Kroll]] &lt;small&gt;as [[Sylvester Stallone]]&lt;/small&gt;&lt;br&gt;[[Paul F. Tompkins]] &lt;small&gt; as [[Len Wiseman]]&lt;/small&gt;
2229 |[[Andy Samberg]]&lt;br&gt;[[Claudia O'Doherty]]&lt;br&gt;[[Nick Kroll]] &lt;small&gt;as R. Schrift, the Mailer Daemon, Fabrice Fabrice, himself, [[Sylvester Stallone]], El Chupacabra, [[Mrs. Doubtfire]], [[Bill Maher]], Creaky Gate, and Gil Faizon&lt;/small&gt;
2230 | [[Sylvester Stallone]]
2231 Fiore shot 17 episodes of the television series ''[[Tracey Takes On...]].'' His first major motion picture credit as cinematographer was the [[Sylvester Stallone]] vehicle ''[[Get Carter (2000 film)|Get Carter]]'' (2000). He established a partnership with director [[Antoine Fuqua]], beginning with ''[[Training Day]]'' (2001), and has shot six more of the director's films since.
2232 |Guests=[[Charles Cozart]], [[Jan Hammer]], [[Sylvester Stallone]]
2233 |Guests=[[Clyde Drexler]], [[Lisa Kudrow]], [[Sylvester Stallone]]
2234 * [[Sylvester Stallone]], [[James Cameron]], and [[Kevin Jarre]] - ''[[Rambo: First Blood Part II]]'' (1985)
2235 However, their efforts to break into the mainstream with [[mockbuster]]s such as ''[[Freefall (1994 film)|Freefall]]'' starring [[Eric Roberts]], [[Jeff Fahey]] and [[Pamela Gidley]], created to capitalise on the success of [[Sylvester Stallone]]'s ''[[Cliffhanger (film)|Cliffhanger]]'' failed to make much of an impact. ''[[Freefall (1994 film)|Freefall]]'' ended up going [[direct-to-video]].&lt;ref&gt;{{cite web|url=https://ultimateactionmovies.com/eric-roberts-freefall-review/|title=ERIC ROBERTS TRIES TO OUT CLIFFHANGER STALLONE IN 'FREEFALL' (1994)|website=Ultimate Action Movies|date=13 February 2020 }}&lt;/ref&gt;
2236 | &lt;!-- 2023.09.23 --&gt;''[[Expend4bles]]'' || || &lt;ref&gt;{{cite web|url=https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video|title=Expendables 4: Sylvester Stallone Looks Thoroughly Badass In New Set Video|first=Erik|last=Swann|work=[[CinemaBlend]] |date=October 12, 2021|access-date=October 14, 2021|url-status=live|archive-date=October 14, 2021|archive-url=https://web.archive.org/web/20211014065910/https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video}}&lt;/ref&gt;
2237 | * [[Britney Spears]] ||   || * [[Sylvester Stallone]] ||   || * [[Patrick Stewart]]
2238 Dugan has overseen numerous successful [[social media marketing]] campaigns and has cultivated relationships with such brands as [[Rogue Pictures]], Fox, Sony Pictures, FedEx, Heineken, Universal Studios, [[Fanista]] and more. He produced and directed online spots featuring such talent as [[Jackson Browne]], [[Sylvester Stallone]] and [[Maná]] for the &quot;BURMA, It Can't Wait&quot; campaign, as well as produced the groundbreaking HIV prevention campaign entitled &quot;In The Moment&quot;.&lt;ref&gt;{{cite web |first=Nikki |last=Finke |title=SAG Holds New Media Do-It-Yourself |url=https://www.deadline.com/2009/02/sag-holds-new-media-do-it-yourself/ |publisher=Deadline.com |date=2009-02-17 |quote=Andy Dugan, Emmy Award-winning producer of LAGLC's groundbreaking social health Web serial In the Moment... }}&lt;/ref&gt;
2239 He was the engineer and pilot for the 2010 Sylvester Stallone film ''[[The Expendables (2010 film)|The Expendables]]''.&lt;ref name=&quot;expendables&quot;&gt;[https://www.imdb.com/title/tt1320253/combined/ &quot;The Expendables, 2010&quot;], IMDb.com, Inc.&lt;/ref&gt;
2240 In 2014, Gisella Marengo embarked on her journey as a film producer. She produced several projects, including &quot;[[The Humbling (film)|The Humbling]]&quot; starring [[Al Pacino]] and directed by [[Barry Levinson]],&lt;ref name=&quot;:0&quot; /&gt; and the American drama &quot;[[Blackbird (2019 film)|Blackbird]],&quot; featuring [[Susan Sarandon]] and [[Kate Winslet]], directed by [[Roger Michell]].&lt;ref name=&quot;:0&quot; /&gt; She also worked on action movies such as &quot;[[Angel Has Fallen]]&quot; with [[Gerard Butler|Gerald Butler]] and [[Morgan Freeman]]&lt;ref name=&quot;:0&quot; /&gt; &quot;[[Hunter Killer (film)|Hunter Killer]]&quot; starring [[Gerard Butler|Gerald Butler]] and [[Gary Oldman]], and &quot;Security&quot; featuring [[Antonio Banderas]].&lt;ref name=&quot;:0&quot; /&gt; Among her recent productions, in 2020, Gisella Marengo produced  the documentary &quot;[[Francesco (2020 film)|Francesco]],&quot; directed by Oscar-nominated [[Evgeny Afineevsky]].&lt;ref name=&quot;:0&quot; /&gt; The film premiered at the [[Rome Film Festival]]&lt;ref&gt;{{Cite web |date=2021-03-26 |title=The film &quot;Francesco&quot; on Discovery+ - Vatican News |url=https://www.vaticannews.va/en/vatican-city/news/2021-03/documentary-film-francesco-afineevsky-pope-francis-streaming.html |access-date=2024-02-21 |website=www.vaticannews.va |language=en}}&lt;/ref&gt; and was awarded with the 18th Kinéo Prize.&lt;ref&gt;{{Cite web |date=2020-10-17 |title=Festa Roma: premio Kineo a Afineevsky per docufilm sul Papa - Cinema - Ansa.it |url=https://www.ansa.it/sito/notizie/cultura/cinema/2020/10/17/festa-roma-premio-kineo-a-afineevsky-per-docufilm-sul-papa_ab6a123e-e79b-4b03-8f9e-1cb3e1bcb432.html |access-date=2024-02-21 |website=Agenzia ANSA |language=it}}&lt;/ref&gt; The following year, she was the producer behind the highly acclaimed and renowned documentary &quot;[[Ennio (film)|Ennio]],&quot; which delved into the life of the famous Italian Oscar-winning film composer, [[Ennio Morricone]].&lt;ref name=&quot;:1&quot;&gt;{{Cite web |title=Prime Video: ENNIO THE MAESTRO |url=https://www.primevideo.com/detail/ENNIO-THE-MAESTRO/0O398U97308YFWR5YB0LSSLTUZ |access-date=2024-02-21 |website=www.primevideo.com |language=en-us}}&lt;/ref&gt; Directed by Oscar winning director [[Giuseppe Tornatore]], the documentary features interviews with  world-class artists, including [[Clint Eastwood]], [[Quentin Tarantino]], [[Hans Zimmer]], and [[Bruce Springsteen]].&lt;ref name=&quot;:1&quot; /&gt; It made its debut at the [[Venice Film Festival|Venice International Film Festival]]&lt;ref&gt;{{Cite web |date=2021-08-10 |title=Biennale Cinema 2021 {{!}} Ennio by Giuseppe Tornatore screens Out of Competition at the 78th Festival |url=https://www.labiennale.org/en/news/ennio-giuseppe-tornatore-screens-out-competition-78th-festival |access-date=2024-02-21 |website=La Biennale di Venezia |language=en}}&lt;/ref&gt; and it was awarded with the prestigious [[David di Donatello|David di Donatello Award]] for Best Documentary and the Starlight International Film Award.&lt;ref&gt;{{Cite web |date=2022-05-04 |title=Ai David di Donatello vola 'E' stata la mano di Dio' di Paolo Sorrentino - Notizie - Ansa.it
```



```
|url=https://www.ansa.it/sito/notizie/cultura/2022/05/04/ai-david-di-donatello-vola-e-stata-la-mano-di-dio-di-paolo-sorrentino_63e3d407-8794-42a3-9480-3cd4a683d057.html |access-date=2024-02-21 |website=Agenzia ANSA |language=it}}&lt;/ref&gt; In 2023, Marengo co-produced the action-packed &quot;[[Expend4bles]],&quot; featuring actors like [[Megan Fox]], [[Sylvester Stallone]], and [[Jason Statham]].&lt;ref name=&quot;:0&quot; /&gt;
2241 [[Category:Images of Sylvester Stallone]]</text>
2242 :Europeans call them American unless a nation wants to claim one of the them as &quot;their own&quot; such as Ireland did with John F. Kennedy and the Italians do with Sylvester Stallone and Robert De Niro. In my youth, I had a French friend who excitedly informed her dad that I wasn't a &quot;real&quot; American on account that I had a French great-grandmother!--[[User:Jeanne boleyn|Jeanne Boleyn]] ([[User talk:Jeanne boleyn|talk]]) 07:54, 13 June 2010 (UTC)
2243 Powell's acting career began while working with [[Antonio Banderas]] and [[Sylvester Stallone]] in ''[[Spy Kids 3-D: Game Over]]''.&lt;ref&gt;{{Cite web |last=Hogan |first=Kate |date=2024-05-02 |title=Remember Glen Powell's Part in 'Spy Kids 3-D'? He Calls Filming 'One of the Best Days' of His Life |url=https://people.com/glen-powell-talks-filming-spy-kids-3d-game-over-8652337 |access-date=2024-05-24 |website=Peoplemag |language=en |archive-date=May 24, 2024 |archive-url=https://web.archive.org/web/20240524021408/https://people.com/glen-powell-talks-filming-spy-kids-3d-game-over-8652337 |url-status=live }}&lt;/ref&gt; In 2007, before his first year of college, Powell landed a role in ''[[The Great Debaters]]'', directed by and starring [[Denzel Washington]].&lt;ref&gt;{{cite web |last1=Eisenberg |first1=Eric |date=August 12, 2013 |title=The Expendables 3 Adds Newcomer Glen Powell |url=http://www.cinemablend.com/new/Expendables-3-Adds-Newcomer-Glen-Powell-38964.html |url-status=live |archive-url=https://web.archive.org/web/20170224095820/http://www.cinemablend.com/new/Expendables-3-Adds-Newcomer-Glen-Powell-38964.html |archive-date=February 24, 2017 |access-date=November 20, 2015 |website=Cinema Blend}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Holloway |first=Diane |date=October 12, 2007 |title=Austin teens compete on 'Endurance: Fiji' |url=http://www.austin360.com/blogs/content/shared-gen/blogs/austin/tvblog/entries/2007/10/12/austin_teens_compete_on_endurance_fiji.html |url-status=dead |archive-url=https://web.archive.org/web/20160529154925/http://www.austin360.com/blogs/content/shared-gen/blogs/austin/tvblog/entries/2007/10/12/austin_teens_compete_on_endurance_fiji.html/ |archive-date=May 29, 2016 |access-date=June 20, 2010 |publisher=Austin360.com}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last=Barnes |first=Michael |date=December 24, 2007 |title=Glen Powell Jr. picks up debate tips from Denzel Washington |url=http://mo.austin360.com/blogs/content/shared-gen/blogs/austin/outandabout/entries/2007/12/24/glen_powell_jr.html |url-status=dead |archive-url=https://web.archive.org/web/20110707190700/http://mo.austin360.com/blogs/content/shared-gen/blogs/austin/outandabout/entries/2007/12/24/glen_powell_jr.html |archive-date=July 7, 2011 |access-date=June 20, 2010 |publisher=Austin360.com |df=mdy-all}}&lt;/ref&gt; Washington introduced him to agent [[Ed Limato]], who encouraged Powell to move to Los Angeles. After moving to the city, Powell stayed with a friend of the family, and he later described struggling during this period to get roles, including failed auditions for ''[[Friday Night Lights (TV series)|Friday Night Lights]],'' ''[[Cowboys &amp; Aliens]],'' and ''[[The Longest Ride (film)|The Longest Ride]]''.&lt;ref name=&quot;:42&quot;&gt;{{Cite web |last=Rose |first=Lacey |date=2024-05-22 |title=Glen Powell Finally Conquered Hollywood. So Why Is He Leaving? |url=https://www.hollywoodreporter.com/movies/movie-features/glen-powell-interview-hit-man-movie-texas-move-1235905159/ |access-date=2024-05-24 |website=The Hollywood Reporter |language=en-US |archive-date=May 24, 2024 |archive-url=https://web.archive.org/web/20240524152630/https://www.hollywoodreporter.com/movies/movie-features/glen-powell-interview-hit-man-movie-texas-move-1235905159/ |url-status=live }}&lt;/ref&gt;&lt;ref name=&quot;:0&quot;&gt;{{Cite web |last=Hartman |first=Eva |date=2024-05-28 |title=Glen Powell initially turned down his 'Top Gun' role - so Tom Cruise rewrote it for him |url=https://www.latimes.com/entertainment-arts/movies/story/2024-05-28/glen-powell-tom-cruise-top-gun-rewrite |access-date=2024-05-29 |website=[[Los Angeles Times]] |language=en-US |archive-date=May 29, 2024 |archive-url=https://web.archive.org/web/20240529075116/https://www.latimes.com/entertainment-arts/movies/story/2024-05-28/glen-powell-tom-cruise-top-gun-rewrite |url-status=live }}&lt;/ref&gt; However, he saw some success with small credits in television series, including ''[[Into the West (miniseries)|Into the West]]'', ''[[Jack &amp; Bobby]]'', ''[[CSI: Miami]]'', ''[[NCIS (TV series)|NCIS]]'', ''[[Without a Trace]]'', ''[[Rizzoli &amp; Isles]]'', and ''[[The Lying Game (TV series)|The Lying Game]].'' He also had minor parts in ''[[The Dark Knight Rises]]'' and ''[[Stuck in Love (film)|Stuck in Love]].''&lt;ref&gt;{{cite web |last1=Valby |first1=Karen |title=Game Changers 2015: Glen Powell |url=http://www.mensfitness.com/life/entertainment/game-changers-glen-powell |url-status=live |archive-url=https://web.archive.org/web/20170206215046/http://www.mensfitness.com/life/entertainment/game-changers-glen-powell |archive-date=February 6, 2017 |access-date=November 20, 2015 |website=Men's Fitness}}&lt;/ref&gt;
2244 Powell began receiving larger roles in feature films around 2014. He played a hacker in ''[[The Expendables 3]]'', an action movie that starred several well-known stars and that reunited Powell with Sylvester Stallone and Antonio Banderas. Powell later recounted asking Stallone for advice while filming on how to succeed in Hollywood.&lt;ref&gt;{{Cite web |last=Gross |first=Joe |title=Austin-raised actor Glen Powell talks 'The Expendables 3' |url=https://www.statesman.com/story/news/2014/08/14/austin-raised-actor-glen-powell-talks-the-expendables-3/10137881007/ |access-date=2024-05-24 |website=Austin American-Statesman |language=en-US |archive-date=October 4, 2022 |archive-url=https://web.archive.org/web/20221004135911/https://www.statesman.com/story/news/2014/08/14/austin-raised-actor-glen-powell-talks-the-expendables-3/10137881007/ |url-status=live }}&lt;/ref&gt; Powell next took minor roles in the comedies ''[[Sex Ed (film)|Sex Ed]]'' and ''[[Ride Along 2]]''. During this period, he also had a main role in the first season of the television show ''[[Scream Queens (2015 TV series)|Scream Queens]]'' and a recurring role in the second season.&lt;ref&gt;{{Cite web |last=Wagmeister |first=Elizabeth |date=2016-03-18 |title=Glen Powell Doesn't Even Know His 'Scream Queens' Season 2 Storyline |url=https://variety.com/2016/tv/news/scream-queens-season-2-characters-glen-powell-chad-radwell-1201733304/ |access-date=2024-05-24 |website=Variety |language=en-US |archive-date=May 27, 2024 |archive-url=https://web.archive.org/web/20240527165326/https://variety.com/2016/tv/news/scream-queens-season-2-characters-glen-powell-chad-radwell-1201733304/ |url-status=live }}&lt;/ref&gt; In March 2016, he co-starred as Finnegan in ''[[Everybody Wants Some!! (film)|Everybody Wants Some!!]]'', [[Richard Linklater]]'s spiritual sequel to ''[[Dazed and Confused (film)|Dazed &amp; Confused]]'', which was filmed in Austin, Texas and released by Paramount.&lt;ref&gt;{{cite magazine |last1=Blistein |first1=Jon |date=November 18, 2015 |title=Richard Linklater's 'Everybody Wants Some' to Open SXSW |url=https://www.rollingstone.com/movies/news/richard-linklaters-everybody-wants-some-to-open-sxsw-20151117 |url-status=live |archive-url=https://web.archive.org/web/20151119143120/http://www.rollingstone.com/movies/news/richard-linklaters-everybody-wants-some-to-open-sxsw-20151117 |archive-date=November 19, 2015 |access-date=November 20, 2015 |magazine=Rolling Stone}}&lt;/ref&gt; Later that year, Powell played astronaut [[John Glenn]] in the biographical drama ''[[Hidden Figures]]''.&lt;ref&gt;{{Cite web |last=Ehrbar |first=Ned |date=2017-02-28 |title=Oscars 2017: &quot;Hidden Figures&quot; star Glen Powell salutes Bill Paxton on the red carpet - CBS News |url=https://www.cbsnews.com/news/oscars-2017-hidden-figures-glen-powell-bill-paxton/ |access-date=2024-05-24 |website=www.cbsnews.com |language=en-US |archive-date=May 27, 2024 |archive-url=https://web.archive.org/web/20240527165332/https://www.cbsnews.com/news/oscars-2017-hidden-figures-glen-powell-bill-paxton/ |url-status=live }}&lt;/ref&gt; In 2017, he had a minor role as Sergeant Dylan Chutsky in the Netflix film ''[[Sand Castle (film)|Sand Castle]]''.&lt;ref&gt;{{Cite web |last=N'Duka |first=Amanda |date=2015-10-14 |title=Glen Powell Enlists In 'Sand Castle'; Diane Guerrero Joins 'The Godmother' |url=https://deadline.com/2015/10/glen-powell-sand-castle-diane-guerrero-the-godmother-1201581543/ |access-date=2024-05-24 |website=Deadline |language=en-US |archive-date=October 1, 2020 |archive-url=https://web.archive.org/web/20201001184236/https://deadline.com/2015/10/glen-powell-sand-castle-diane-guerrero-the-godmother-1201581543/ |url-status=live }}&lt;/ref&gt; He then starred in ''[[The Guernsey Literary and Potato Peel Pie Society (film)|The Guernsey Literary and Potato Peel Pie Society]]''.&lt;ref&gt;{{Cite web |last=Windsor |first=Harry |date=2018-04-12 |title='The Guernsey Literary and Potato Peel Pie Society': Film Review |url=https://www.hollywoodreporter.com/movies/movie-reviews/guernsey-literary-potato-peel-pie-society-review-1102389/ |access-date=2024-08-01 |website=The Hollywood Reporter |language=en-US}}&lt;/ref&gt;
2245 After first working as an accountant, originally in a Manhattan firm, later as a solo accountant, then in an investment firm founded with his brother Warren Starr, an attorney, Starr finally formed his own investment company, Starr &amp; Company, LLC (Starrco). Starr gained wealthy and well-known clients, business associates and friends including Blackstone founder [[Peter George Peterson|Pete Peterson]], restaurateur Shelly 'Shelly's' Fireman, and a ''Who's-Who'' of many Hollywood stars including [[Al Pacino]], [[Natalie Portman]], [[Martin Scorsese]], [[Carly Simon]], [[Wesley Snipes]], [[Sylvester Stallone]], and [[Uma Thurman]].&lt;ref name=&quot;VanityFair&quot;/&gt; Minimizing the scope of his thefts, he has been called a 'mini-[[Bernard Madoff|Madoff]]'&lt;ref&gt;{{cite news| first = Dylan| last = Stableford| url = http://www.thewrap.com/movies/article/ken-starr-hollywood-mini-madoff-gets-7-5-year-jail-sentence-25208/| title = Ken Starr, Hollywood's Mini-Madoff, Gets 7.5-Year Jail Sentence| newspaper = [[TheWrap]]| publisher = The Wrap News Inc.| location = Los Angeles, California| date = March 3, 2011| accessdate = January 2, 2015| archive-url = https://web.archive.org/web/20150103005750/http://www.thewrap.com/movies/article/ken-starr-hollywood-mini-madoff-gets-7-5-year-jail-sentence-25208/| archive-date = January 3, 2015| url-status = live}}&lt;/ref&gt;&lt;ref&gt;{{cite news| first =Dan| last =Primack| title =&quot;Letters&quot; in support of a mini-Madoff| url =http://fortune.com/2011/01/19/letters-in-support-of-a-mini-madoff/| newspaper =[[Fortune (magazine)|Fortune]]| publisher =[[Meredith Corporation]]| location =New York City| date =January 19, 2011| accessdate =January 2, 2015| archive-url =https://web.archive.org/web/20150103005700/http://fortune.com/2011/01/19/letters-in-support-of-a-mini-madoff/| archive-date =January 3, 2015| url-status =live}}&lt;/ref&gt;&lt;ref&gt;{{cite news|first=Andrew Ross|last=Sorkin|title=Adviser to Stars Named in Madoff-Like Fraud|url=https://dealbook.nytimes.com/2010/05/28/adviser-to-stars-named-in-madoff-like-fraud/|newspaper=[[The New York Times]]|location=New York City|date=May 28, 2010|accessdate=January 2, 2015|quote=But when they arrested him on Thursday, federal prosecutors described him as something else: a mini-Madoff who diverted $30 million of his clients' money to buy himself a sprawling Upper East Side condo complete with an indoor swimming pool and a 1,500-square-foot garden.|archive-url=https://web.archive.org/web/20150103005809/http://dealbook.nytimes.com/2010/05/28/adviser-to-stars-named-in-madoff-like-fraud/|archive-date=January 3, 2015|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite news|first=Robert|last=Schmidt|title=Wayne McLeod: The Life and Death of a Mini-Madoff|url=http://www.businessweek.com/magazine/content/10_37/b4194050927939.htm|newspaper=[[Bloomberg Businessweek]]|publisher=[[Bloomberg L.P.]]|location=New York City|date=September 2, 2010|accessdate=January 2, 2015|quote=The financial crisis also led to the arrests of a string of &quot;mini-Madoffs,&quot; whose crimes, while not as massive, were just as brazen. They include New York money manager Kenneth Starr, arrested in May and charged with stealing at least $59 million from movie stars and other celebrities.|archive-url=https://web.archive.org/web/20140824104439/http://www.businessweek.com/magazine/content/10_37/b4194050927939.htm|archive-date=August 24, 2014|url-status=dead}}&lt;/ref&gt; for the kind of clients he attracted, the breach of trust, and his notoriety.
2246 Starr's financial and legal problems began to surface in the late-1990s after he was sued by [[Sylvester Stallone]] in connection with the [[Planet Hollywood]] restaurant chain and Starr's alleged role in Stallone's $10 million loss. As his financial world began to crumble, Starr's then-wife, Marisa Vucci Starr, resigned as Starrco's office manager.&lt;ref name=&quot;VanityFair&quot;/&gt; This left only Starr's son, Ron Starr (identified though not a subject of  the indictment as Ron''Star Bristol''), also an attorney and then co-Chief Compliance Officer, as the only family member left at the company. At the time of Starr's arrest, Ron Starr, along with the company's other co-Chief Compliance Officer, were-with the assistance of outside counsel-conducting an internal investigation into Starr's conduct, and had already forced Starr to surrender signatory authority over all client accounts.  Ron Starr then proceeded to assist the authorities in his father's prosecution and continues to work as a financial adviser in Manhattan.
2247 {{cquote|We wanted to have entertainers who are truly public favorites, like [[Arnold Schwarzenegger]] or [[Sylvester Stallone]] or [[Jean-Claude Van Damme]], people who do big box office and big business in home video, but are not in the foreseeable future likely to be nominated for an Academy Award.|Brian Woods&lt;ref name=more/&gt;}}
2248 *[[Sylvester Stallone]]{{spaced ndash}} actor
2249     <title>Category:Films directed by Sylvester Stallone</title>
2250       <text bytes="165" sha1="1kodsgsopg8bg19twdz0pseguzitmt4" xml:space="preserve">{{Cat more|Sylvester Stallone}}
2251 [[Category:Works by Sylvester Stallone|Films directed]]</text>
2252 * 1990BLACK AND BLOO, Sylvester Stallone and William Quigley,  April, Los Angeles, CA
2253 :From what American movies have taught me, it will not be fixed until Sylvestor Stallone goes down there and sprays it with several thousand bullets from his over-sized machine gun. [[Special:Contributions/92.24.181.157|92.24.181.157]] ([[User talk:92.24.181.157|talk]]) 11:33, 9 July 2010 (UTC)
2254 Rosado appeared in the boxing film ''[[Creed (film)|Creed]]'' (2015), starring [[Michael B. Jordan]] and [[Sylvester Stallone]].&lt;ref&gt;[http://articles.philly.com/2014-12-09/news/
```



```
     56844325_1_gabriel-rosado-left-eye-fighter Eye problem plaguing Gabriel Rosado] published by Matt Breen, Inquirer Staff Writer on December 9, 2014&lt;/ref&gt;&lt;ref&gt;[http://uproxx.com/movies/2015/06/michael-b-jordan-creed-trailer/ The Trailer For The ‘Rocky’ Spinoff ‘Creed’ Has Arrived]
2255 # 6,363 [&lt;.01%]: [[Sylvester Stallone]]
2256 | ShortSummary    = Items appraised include two lithographs by [[Norman Rockwell]]; a vintage Relax-IT massager; a block of plaster with [[Sylvester Stallone]]'s handprints; and a vintage ''[[Sky King]]'' pedal plane.
2257 | ShortSummary    = Continuing their Michigan trip from the previous episode, Rick and Chumlee check out a [[116th Infantry Brigade Combat Team|116th Infantry Brigade]] sign; a 1940s [[Douglas Aircraft Company]] rowboat; an Oilzum Motor Oil sign; a [[Ford FE engine|Ford FE 427 engine]]; a 1950s [[RC Cola]] cooler; and a World War I experimental Model 8 helmet, which is authenticated by a [[Ford Motor Company]] archivist. Back in Las Vegas, items appraised include a Japanese torpedo [[gyroscope]] from World War II; a ''[[Rocky III]]'' statue signed by [[Sylvester Stallone]]; a tie clasp said to be owned by [[Richard Nixon]]; an 1859 etching by [[James McNeill Whistler]]; a [[Carnegie Steel Company]] scale model of a [[fishplate]]; and a letter written by [[Ted Bundy]].
2258 | ShortSummary    = Items appraised include two signed letters by [[Albert Einstein]]; a 1953 shuffleboard/bowling arcade game; a replica [[Thor (Marvel Comics)|Thor]] hammer and helmet; a [[McDonald's]] [[Big Mac]] jacket; a 1910 baseball stitch machine; an unopened box of ''[[Pokémon]]'' cards; two ancient Roman coins; a 1980s portrait of [[Sylvester Stallone]]; a decommissioned [[Thompson submachine gun]]; and an M1 cutaway gun.
2259 * ''[[Rambo (2008 film)|Rambo]]'' (2008) - starring [[Sylvester Stallone]]
2260 &quot;'''Sinner's Prayer'''&quot;  is a song by American rock musician [[Sully Erna]]. It is the [[lead single]] from his 2010 debut album ''[[Avalon (Sully Erna album)|Avalon]]''. The song was originally written for Sylvester Stallone's blockbuster ''[[The Expendables (2010 film)|The Expendables]]'' but later taken off the film and soundtrack during post-production. However, the song is reinstated back in the film in the extended director's cut during the films new opening credits sequence.
2261 On September 25, Sully Erna told the Artisan News Service in a video interview that he was approached to record &quot;Sinner's Prayer&quot; for this year's [[Sylvester Stallone]] blockbuster &quot;[[The Expendables (2010 film)|The Expendables]]&quot;, which will also feature [[Jason Statham]], [[Jet Li]], [[Mickey Rourke]] and [[Randy Couture]].&lt;ref&gt;[http://www.metalrage.com/news/52034/godsmack_singer_hooks_up_with_sylvester_stallone_for_movie_soundtrack.html/ &quot;Sully Erna Working on The Expendables soundtrack&quot;] ''Metalrage.com''. Retrieved on November 19, 2009.&lt;/ref&gt;
2262 Abutbul's first role in a big popular film was in 1986 when he was cast in the Israeli movie ''Shtei Etzbaot Mi'Tzidon''&lt;ref&gt;{{cite web|title=Shtei Etzbaot Mi'Tzidon (Ricochets)|url=http://www.rottentomatoes.com/m/shtei-etzbaot-mitzidon-ricochets/|publisher=[[Rotten Tomatoes]]|access-date=18 January 2016|date=8 September 1986}}&lt;/ref&gt; (''Two Fingers from Sidon''), a film which took place in the [[1982 Lebanon War]]. For his performance in this film Abutbul was later awarded the &quot;Best Actor&quot; award at the Jerusalem Film Festival. That year he also played in the American-Israeli co-production ''[[Every Time We Say Goodbye (film)|Every Time We Say Goodbye]]'' in which he played alongside [[Gila Almagor]]. In 1987 Abutbul appeared in the Israeli ''Photo Roman''. A year later, Abutbul played in the Israeli film ''Makom L'Yad Hayam'' alongside Anat Tzachor and in the American film ''[[Rambo III]]'' alongside [[Sylvester Stallone]].
2263 {{cquote|A few months ago we got a phone call from a very interesting individual. It's not everyday that Rambo and Rocky call you at the same time. [[Sylvester Stallone]] wanted a song for his new movie, called ''The Expendables''. He said write me something that will personify not only the South, but what rock truly is.&lt;ref&gt;{{cite web|author= |url=https://www.youtube.com/watch?v=8HJJfFNtfLs  |archive-url=https://ghostarchive.org/varchive/youtube/20211221/8HJJfFNtfLs |archive-date=2021-12-21 |url-status=live|title=Diamond Eyes **Boom Lay Boom** SHINEDOWN -LIVE- 7.7.10 |publisher=YouTube |date= |accessdate=February 13, 2012}}{{cbignore}}&lt;/ref&gt;}}
2264  |ShortSummary=Web Redemption for [[David After Dentist]] featuring comedian [[Jimmy Pardo]]. Daniel arm-wrestles [[David Mendenhall]], who played [[Sylvester Stallone]]'s son Michael Hawk in the 1987 film ''[[Over the Top (1987 film)|Over the Top]]''.&lt;ref&gt;{{cite web|url=http://tosh.comedycentral.com/blog/2010/01/13/in-this-weeks-show-12/ |title=In This Week's Show |publisher=Comedy Central |date=January 13, 2010 |access-date=November 9, 2013}}&lt;/ref&gt; Daniel teases the audience with a blacked version of &quot;the blob&quot; which is a result of six years of injecting [[silicone]] into his penis.&lt;ref&gt;{{cite web |url=http://www.siliconpenis.com/ |title=Silicon Penis silicone penile implants 6 years injecting penis with silicone |publisher=Siliconpenis.com |access-date=November 9, 2013 |archive-url=https://web.archive.org/web/20131109210850/http://www.siliconpenis.com/ |archive-date=November 9, 2013 |url-status=dead }}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://tosh.comedycentral.com/blog/2010/01/13/the-blob/ |title=The Blob &amp;#124; Tosh.0 Blog |publisher=Comedy Central |date=January 13, 2010 |access-date=November 9, 2013}}&lt;/ref&gt; Daniel discusses selling a [[cardigan (sweater)|cardigan]] sweater on [[eBay]] for $523, meaning his celebrity is $334 more valuable.&lt;ref&gt;{{cite web|url=http://tosh.comedycentral.com/video-clips/sweater-sale |title=Sweater Sale - Video Clip |publisher=Comedy Central |date=January 13, 2010 |access-date=November 9, 2013}}&lt;/ref&gt; Daniel also attempts to break the world record for most balls rolled over in sequence, in the new segment called, &quot;I'm Better Than You. Na Na Na Boo-boo, Stick Your Head in Doo-doo.&quot;,&lt;ref&gt;{{cite web|url=http://tosh.comedycentral.com/video-clips/w9wyvo/tosh-is-better-than-you---six-balls |title=Tosh Is Better Than You - Six Balls - Video Clip |publisher=Comedy Central |date=January 13, 2010 |access-date=September 20, 2015}}&lt;/ref&gt; and plays &quot;Is it racist?&quot; in a video of a boy spelling [[Negus]] in a [[spelling bee]].&lt;ref&gt;{{cite web|url=http://tosh.comedycentral.com/blog/2010/01/13/is-it-racist-2/ |title=Is It Racist? |publisher=Comedy Central |date=January 13, 2010 |access-date=November 9, 2013}}&lt;/ref&gt;
2265  | footer    = George W. Bush, Sylvester Stallone and Fred Dryer, born July 6, 1946
2266  | image2    = Sylvester Stallone.jpg
2267 **[[Sylvester Stallone|Sylvester Enzio Stallone]], American film actor (''Rocky'' and ''Rambo''), in New York City to Frank Stallone Sr. and [[Jackie Stallone]]; he was injured during the delivery, leaving him with a drooping lower lip and what would become a slight, but distinctive, speech impediment.&lt;ref&gt;John Sarkett, ''Extraordinary Comebacks: 201 Inspiring Stories of Courage, Triumph and Success'' p143 (Sourcebooks, Inc., 2007)&lt;/ref&gt;
2268 |description=Sample of &quot;Sinner's Prayer&quot;. The song was originally written for this year's [[Sylvester Stallone]] blockbuster ''[[The Expendables (2010 film)|The Expendables]]''. &quot;Sinner's Prayer&quot; became available to download on August 3, which is the same date it can be heard on local radio stations across the country for the first time.
2269 On September 25, Sully Erna told the Artisan News Service in a video interview that he was approached to record &quot;Sinner's Prayer&quot; for 2010's [[Sylvester Stallone]] film ''[[The Expendables (2010 film)|The Expendables]]'', which will also feature [[Jason Statham]], [[Jet Li]], [[Mickey Rourke]] and [[Randy Couture]].&lt;ref&gt;[http://www.metalrage.com/news/52034/godsmack_singer_hooks_up_with_sylvester_stallone_for_movie_soundtrack.html/ &quot;Sully Erna Working on The Expendables soundtrack&quot;] ''Metalrage.com''. Retrieved on November 19, 2009.&lt;/ref&gt; A few months later, however, Erna stated that the song won't be featured in that movie because Stallone changed the scene the song was attached to.&lt;ref&gt;[http://motleycrue.rockzillatv.com/godsmack-frontman-our-sound-is-much-more-mature-now-apr-12-2010-blabbermouth-net/ &quot;Godsmack Frontman: ‘Our Sound Is Much More Mature Now’ - Apr. 12, 2010 - Blabbermouth.net | Motley Crue News at Rockzilla&quot;] ''Rockzillatv.com''. Retrieved on April 12, 2010.&lt;/ref&gt;
2270 |Guests=[[Sylvester Stallone]], [[Steve Landesberg]], [[Dr. Joyce Brothers]]
2271 |Guests='''[[Helen Reddy]]''' (guest host), [[Sylvester Stallone]], [[Neil Simon]], [[Cleveland Amory]], [[George Miller (comedian)|George Miller]]
2272 |Guests=[[Sylvester Stallone]], Bobby Kelton, [[Thalassa Cruso]]
2273 |Guests=[[Sylvester Stallone]]
2274 |Guests=[[Sylvester Stallone]], [[Steve Martin]]
2275 |Guests='''[[Joan Rivers]]''' (guest host), [[Sylvester Stallone]], [[Charlie Callas]], [[Morgan Fairchild]]
2276 |Guests=[[Sylvester Stallone]], [[A. Whitney Brown]]
2277 |Guests=[[Sylvester Stallone]], [[Max Alexander (comedian)|Max Alexander]], [[Orson Bean]]
2278 |Guests=[[Sylvester Stallone]]
2279 | [[Sylvester Stallone]]
2280 Sainz became a co-host for the [[Netflix]] reality show ''[[Ultimate Beastmaster]]'' in 2016.&lt;ref&gt;{{cite web|url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|title=Netflix Orders Global Competition Series ‘Ultimate Beastmaster’ from Sylvester Stallone, Dave Broome|last=Prudon|first=Laura|work=Variety|date=May 9, 2016|access-date=October 5, 2016}}&lt;/ref&gt; She hosted seasons 1 and 3 for Mexico.
2281 16. Surely Mr. John Cornell (producer, actor and writer of Paul Hogan projects) who references and links his businessman's website on the first paragraph of his Main page ‘John Cornell’, or Mr. Sylvester Stallone (ditto), with his informative tables, cannot be seen as self promotional?
2282 | [[Sylvester Stallone]], [[Gabourey Sidibe]]
2283 | [[Sylvester Stallone]], [[Chris Brown (American singer)|Chris Brown]] &amp; [[Bow Wow (rapper)|Bow Wow]]
2284 On 2 November 2010, [[Lions Gate Entertainment]] secured the North American distribution rights to ''Dredd''.&lt;ref name=&quot;VarietyNov2010&quot; /&gt; Headey joined the cast as drug-dealer Ma-Ma in January 2011.&lt;ref name=&quot;2011JanPlaylist&quot; /&gt; ''Judge Dredd'' creator [[John Wagner]] acted as a consultant on the film.&lt;ref name=&quot;TheGuardian&quot; /&gt; In 2012, he confirmed that it was a new adaptation of the comic material and was not a remake of the 1995 adaptation ''[[Judge Dredd (film)|Judge Dredd]]'', which starred [[Sylvester Stallone]].&lt;ref name=&quot;2012Empire&quot; /&gt;
2285 Sikander also appeared in the 2007 film ''[[Rocky Balboa (film)|Rocky Balboa]]''&lt;ref name=&quot;IMDb&quot;/&gt; directed by [[Sylvester Stallone]].&lt;ref name=&quot;bio&quot;/&gt;
2286 His paintings hang in such notable buildings as the Queen Sofia Art Museum in [[Madrid]], the Beaux Arts Museum in [[Brussels]], the Contemporary Art Museum in [[Vienna]], the Pretoria Art Museum in [[South Africa]]&lt;ref name=&quot;Pretoria&quot; /&gt; and the [[São Paulo Museum of Modern Art]] in Brazil.&lt;ref name=&quot;AB&quot;/&gt; His works have also been acquired by various celebrities and institutions, such as King [[Juan Carlos I]] and Queen [[Sofia of Greece and Denmark|Sofia]] of Spain, [[Princess Christina of the Netherlands]], Prince [[Albert II of Belgium]], Baron [[Benjamin de Rothschild]], the Spanish Ambassador Miguel de Aldasoro, [[Jacqueline Onassis]], [[Paloma Picasso]], [[Yves Saint Laurent (designer)|Yves Saint Laurent]], [[Paco Rabanne]], [[Francis Ford Coppola]], [[Roman Polanski]], [[Sylvester Stallone]], [[Julio Iglesias]], [[Mireille Mathieu]], [[Sara Montiel]], [[Charles Aznavour]], [[Sacha Distel]], Capital Bank founder and president Abel Holtz, Deutsche Bank, Phillips, Boehringer Ingelheim, Bayer, and Sabena Airlines.&lt;ref&gt;&quot;Revista Casa &amp; Estilo Internacional: Arte, Jesus Fuertes, author: Alfonso N. Morán, issue 1, year 1, pages 12,13, Miami, November, 1999&quot;&lt;/ref&gt;&lt;ref&gt;&quot;Gallery Magazine (GSC Media Group):The Blue Painter, author: Pablo J. Gasc, pages 16, 17, Coral Gables, Florida, October, 1997&quot;&lt;/ref&gt;
2287 ** [[Alain Dorval]], French actor, voice of [[Sylvester Stallone]] (d. 2024).
2288 On October 25, 2011, the [[ESPN]] network broadcast ''The Real Rocky'', an original one-hour documentary directed by Feuerzeig as part of its ongoing documentary showcase series, original titled ''[[30 for 30]]'', and more recently re-branded as “ESPN Films.”  The film chronicles the ongoing controversy surrounding the true identity of the eponymous title character from the [[Sylvester Stallone]] ''[[Rocky]]'' franchise, which public opinion and folk wisdom has long held was based on New Jersey heavyweight boxer [[Chuck Wepner]], “The Bayonne Bleeder,” whose chief claim was that he once went 15 rounds with [[Muhammad Ali]], even knocking him down in the ninth round (by, as it was later revealed, stepping on his foot).&lt;ref name=&quot;go.com&quot;&gt;{{cite
```



web|url=https://www.espn.com/espn/espnfilms/story/_/id/6961210/real-rocky|title=ESPN Films: 'The Real Rocky'|date=12 September 2011|access-date=12 July 2018}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.boxingscene.com/media/showphoto.php/photo/6171/title/ali-vswepner/cat/513|archive-url=https://archive.today/20120721065627/http://www.boxingscene.com/media/showphoto.php/photo/6171/title/ali-vswepner/cat/513|url-status=dead|archive-date=July 21, 2012|title=Ali vs. Wepner - Boxing Photos|website=Boxingscene.com|access-date=12 July 2018}}&lt;/ref&gt;

2289 Feuerzeig has completed the screenplay for ''God Bless Tiny Tim'' with filmmaker/playwright [[Julien Nitzberg]]. This portrait of [[Tiny Tim (musician)|the latterday crooner]] and accidental cultural icon is currently in development.&lt;ref&gt;{{cite web|url=http://www.answers.com/topic/tiny-tim-musician|title=Answers - The Most Trusted Place for Answering Life's Questions|website=Answers.com|access-date=12 July 2018}}&lt;/ref&gt; He is also slated to direct ''[[The Bleeder]]'', a biopic of boxer [[Chuck Wepner]], whose life story inspired (and was in turn affected by) the [[Sylvester Stallone]] film ''[[Rocky]]''. Feuerzeig co-wrote the script with [[Jerry Stahl]], and the film is set to star [[Liev Schreiber]] as Wepner.&lt;ref&gt;Rosenberg, Adam; &quot;EXCLUSIVE: Liev Schreiber Wants to Tell Real-Life 'Rocky' Story Next, Director Named&quot; [https://web.archive.org/web/20100303230104/http://moviesblog.mtv.com/2010/02/26/exclusive-liev-schreiber-wants-to-tell-real-life-rocky-story-next-director-named/], Mtv.com, February 10, 2006&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.rsafilms.com/company/rsa-usa/director/jeff-feuerzeig/i-am-chuck-wepner-short-1922|title=RSA Films|website=RSA Films|access-date=12 July 2018|archive-url=https://web.archive.org/web/20120311165529/http://www.rsafilms.com/company/rsa-usa/director/jeff-feuerzeig/i-am-chuck-wepner-short-1922|archive-date=11 March 2012|url-status=dead|df=dmy-all}}&lt;/ref&gt;

2290 *'''Listify, then delete''' - It seems to be undisputed that the award is [[WP:N|notable]] (as a topic) and note''worthy'' (within articles). However, the standard for categorization is much higher: a characteristic must be ''defining'' for the subjects being categorized. Obviously, if a characteristic is not noteworthy, then it is almost certainly not defining; however, the reverse is not true: a characteristic can be noteworthy but still non-defining.&lt;br /&gt;In the case of award categories, consensus is and has been to listify except when the award is among the highest recognitions in the field. The Golden Raspberry Award does not reach that standard, and being a recipient of the Award does not appear to be defining for most of the people being categorized. A quick glance at the first two categories reveals numerous successful/famous (or infamous) actors, musicians, and entertainers who are widely known for their work or (not-so-)personal life: [[Ben Affleck]], [[Kevin Costner]], [[Neil Diamond]], [[Eddie Murphy]], [[Mike Myers (actor)]], [[Prince (musician)]], [[Adam Sandler]], [[Sylvester Stallone]], [[John Travolta]], [[Bruce Willis]], [[Halle Berry]], [[Sandra Bullock]], [[Mariah Carey]], [[Bo Derek]], [[Paris Hilton]], [[Lindsay Lohan]], [[Madonna (entertainer)]], [[Liza Minnelli]], [[Demi Moore]], the [[Spice Girls]], and [[Sharon Stone]]. --'''[[User:Black Falcon|Black Falcon]]''' &lt;sup&gt;([[User talk:Black Falcon|talk]])&lt;/sup&gt; 21:56, 23 November 2010 (UTC)

2291 [[Sylvester Stallone]]{{Dot}}

2292 In 2006, Beristáin became a member of the [[World Boxing Hall of Fame]] as a trainer.&lt;ref&gt;{{Cite web |url=http://www.eastsideboxing.com/news.php?p=9291&amp;more=1 |title=Hall of Fame Trainer Ignacio &quot;Nacho&quot; Beristain Speaks Out! - EastSideBoxing.com |access-date=2010-11-18 |archive-url=https://web.archive.org/web/20120803030825/http://www.eastsideboxing.com/news.php?p=9291&amp;more=1 |archive-date=2012-08-03 |url-status=dead }}&lt;/ref&gt;&lt;ref&gt;[http://www.thesweetscience.com/boxing-article/4480/beristain-humbled-hall-fame-honors/ Beristain humbled by hall of fame honors - TSS.com] {{webarchive|url=https://web.archive.org/web/20100109183257/http://www.thesweetscience.com/boxing-article/4480/beristain-humbled-hall-fame-honors |date=2010-01-09 }}&lt;/ref&gt; Then on December 7, 2010, he was inducted to the [[International Boxing Hall of Fame]], alongside legendary Mexican champion [[Julio César Chávez]], Russian Australian Undisputed Junior Welterweight World Champion, [[Kostya Tszyu]], heavyweight champion [[Mike Tyson]], and actor [[Sylvester Stallone]].&lt;ref&gt;[http://www.ibhof.com/pages/inductionweekend/2011/11announce.html Boxers Chavez, Tszyu and Tyson Elected to Int'l Boxing Hall of Fame - IBHOF.com] {{webarchive|url=https://web.archive.org/web/20110126212008/http://ibhof.com/pages/inductionweekend/2011/11announce.html |date=2011-01-26 }}&lt;/ref&gt;

2293 Through the years, Gladstone licensed works from famous celebrities, artists, photographers and musicians including [[Elvis Presley]] Enterprises, [[Frank Sinatra]] Enterprises, entertainment and pop culture artist [[Joe Petruccio]], The Paradise Collection of Dan Mackin, legendary musician [[Brian Wilson]], [[Jefferson Airplane]]'s [[Marty Balin]], [[The Beatles]] portfolios from The Hulton Archives of London and Mirrorpix, the [[Led Zeppelin]] Collection from [[James Fortune]], [[The Grateful Dead]], [[Bob Marley]], Archives of Peter Simon, and the new [[Michael Ochs]] Collections' classic rock and jazz.&lt;ref name=&quot;Art Meets Entertainment&quot; /&gt;&lt;ref name=&quot;Tennis star&quot; /&gt;&lt;ref&gt;{{Cite news|title=Betty Boop animator Myron Waldman dies|url=https://www.deseret.com/2006/2/7/19936813/betty-boop-animator-myron-waldman-dies|date=February 7, 2006|work=[[Deseret News]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|title=Animator helped bring legends to life|url=https://www.ocregister.com/2006/02/07/animator-helped-bring-legends-to-life/|date=February 7, 2006|work=[[Orange County Register]]}}&lt;/ref&gt;&lt;ref name=&quot;stallone&quot;&gt;{{Cite web|url=https://sylvesterstallone.com/signed-rocky-artwork/|title=Signed Rocky artwork|date=January 20, 2007|access-date=June 7, 2020|website=SylvesterStalone.com}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.tripadvisor.com/ShowTopic-g34345-i53-k2920951-Update_on_Duval_fire_in_todays_paper_Gallery_Montage-Key_West_Florida_Keys_Florida.html|title=Update on Duval fire in todays paper|access-date=June 7, 2020|website=tripadvisor.com|last=Swary|first=Anne-Margaret}}&lt;/ref&gt;&lt;ref name=&quot;ARA2010&quot;&gt;{{Cite press release|title=American Royal Arts Releases New Catalog for 2010|date=January 6, 2010|publisher=[[Cision]]|url=https://www.prweb.com/releases/american-royal-arts/01/prweb3411734.htm|archive-url=https://web.archive.org/web/20100109025858/http://www.prweb.com/releases/american-royal-arts/01/prweb3411734.htm|url-status=dead|archive-date=January 9, 2010}}&lt;/ref&gt;

2294 Some of the entertainment fine art offered by Gladstone and his companies were &quot;The Scottsdale Collection,&quot; art and memorabilia signed by [[Muhammad Ali]]&lt;ref name=&quot;ARA2010&quot; /&gt;&lt;ref&gt;{{Cite web|url=https://jerrygladstonesuccessblog.wordpress.com/2016/06/06/learning-courage-from-the-greatest-muhammad-ali/|title=Learning Courage from The Greatest - Muhammad Ali|date=June 6, 2016|access-date=June 7, 2020}}&lt;/ref&gt; during a witnessed in-person signing, as well as artwork signed by [[Tiger Woods]] and [[Michael Jordan]] through [[Upper Deck]] Authenticated. Through a license agreement with [[Metro-Goldwyn-Mayer|MGM Studios]], he also offered personalized artwork from [[Sylvester Stallone]] celebrating the 30th anniversary of [[Rocky]].&lt;ref name=&quot;stallone&quot; /&gt;

2295 | Actress, Real TV Star and [[Sylvester Stallone]]'s former wife

2296 * [[Sylvester Stallone]]

2297 |Sylvester Stallone

2298 *[[Sylvester Stallone]] (born 1946), film actor, director, producer, and screenwriter (Rocky), maternally Jewish

2299 | [[Sylvester Stallone]]

2300 |[[Sylvester Stallone]] - ''[[Creed (film)|Creed]]''

2301 | ''[[Zookeeper (film)|Zookeeper]]'' || [[Columbia Pictures]] / [[Metro-Goldwyn-Mayer]] || [[Frank Coraci]] (director); [[Nick Bakay]], [[Kevin James]], Rock Reuben, Jay Sherick, David Ronn (screenplay); [[Kevin James]], [[Rosario Dawson]], [[Leslie Bibb]], [[Ken Jeong]], [[Donnie Wahlberg]], [[Joe Rogan]], [[Nat Faxon]], [[Steffiana de la Cruz]], [[Nick Bakay]], [[Jackie Sandler]], [[Nicholas Turturro|Nick Turturro]], [[Thomas Gottschalk]], [[Brandon Keener]], [[Gary Valentine]], [[Nick Nolte]], [[Sylvester Stallone]], [[Adam Sandler]], [[Cher]], [[Judd Apatow]], [[Jon Favreau]], [[Faizon Love]], [[Maya Rudolph]], [[Bas Rutten]], [[Don Rickles]], [[Jim Breuer]], [[Tom Woodruff, Jr.]], [[Bart the Bear 2]], [[Crystal the Monkey]] ||style=&quot;text-align:center;&quot;| &lt;ref&gt;{{cite web |url=https://comingsoon.net/films.php?id=44406 |title=The Zookeeper |website=ComingSoon.net |access-date=2010-05-02}}&lt;/ref&gt;

2302 ** You're right, I was just listening to my ''[[Antipope Sylvester IV]] Greatest Hits'' CD in the car this morning! No one else at [[Sylvester]] is known professionally as &quot;Sylvester&quot;. Should someone one day write an article on &quot;Drinkinstein&quot; it would be categorized as a Sylvester Stallone song and there would be no confusion. [[User:I Want My GayTV|I Want My GayTV]] ([[User talk:I Want My GayTV|talk]]) 01:59, 6 February 2011 (UTC)

2303 In November 2014, Earwolf launched a sister network, Wolfpop, led by [[Paul Scheer]]. Wolfpop's podcast lineup included ''The Black List Table Reads'', ''[[The Canon (podcast)|The Canon]]'', ''Cardboard! With Rich Sommer'', ''Crybabies'', ''[[Denzel Washington Is The Greatest Actor Of All Time Period]]'', ''Get Up on This'', ''Happy Sad Confused'', ''I Was There Too'', ''Maltin on Movies'', ''Nerd Machine's Picking Favorites'', ''Off Camera'', ''OMFG!'', ''Reading Aloud'', ''Rotten Tomatoes'', and ''The Sylvester Stallone Show''. On March 7, 2016, the majority of Wolfpop's programs were folded over into Earwolf.&lt;ref&gt;{{cite web|url=http://forum.earwolf.com/topic/31174-earwolf-and-wolfpop|title=Earwolf and Wolfpop|website=Earwolf Forums}}&lt;/ref&gt;

2304 |[[Sylvester Stallone]]

2305 |Supervisor: fight and training montages&lt;br&gt;Directed by [[Sylvester Stallone]]

2306 |[[Sylvester Stallone]] for ''[[The Expendables (2010 film)|The Expendables]]''

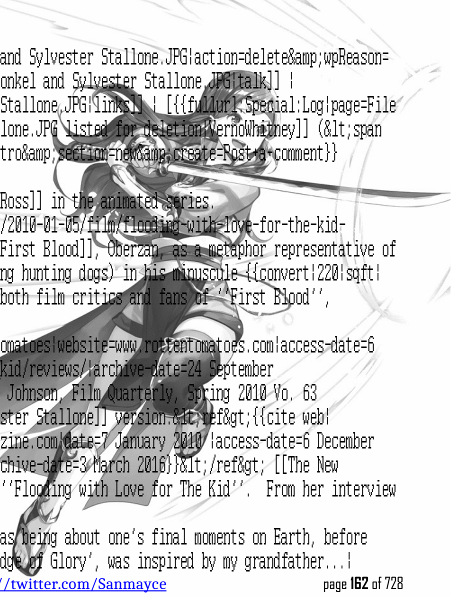
2307 ====[[:File:Bob Van Ronkel and Sylvester Stallone.JPG]]====

2308 :&lt;span class=&quot;plainlinks nourlexpansion lx&quot;&gt;[[:File:Bob Van Ronkel and Sylvester Stallone.JPG]] ([{{fullurl:File:Bob Van Ronkel and Sylvester Stallone.JPG|action=delete&amp;wpReason=%5B%5BWikipedia%3AFiles+for+deletion%2F2011+January+8%23File%3ABob+Van+Ronkel+and+Sylvester+Stallone.JPG%5D%5D}} delete] | [[File talk:Bob Van Ronkel and Sylvester Stallone.JPG|talk]] | [{{fullurl:File:Bob Van Ronkel and Sylvester Stallone.JPG|action=history}} history] | [[Special:WhatLinksHere/File:Bob Van Ronkel and Sylvester Stallone.JPG|links]] | [{{fullurl:Special:Log|page=File%3ABob+Van+Ronkel+and+Sylvester+Stallone.JPG}} logs])&lt;/span&gt; - uploaded by [[User talk:VernoWhitney#File:Bob Van Ronkel and Sylvester Stallone.JPG listed for deletion|VernoWhitney]] (&lt;span class=&quot;plainlinks&quot;&gt;[{{fullurl:User talk:VernoWhitney|action=edit&amp;preload=Template:Fdw_preload&amp;editintro=Template:Fdw_editintro&amp;section=new&amp;create=Post+a+comment}} notify]&lt;/span&gt; | [[Special:Contributions/VernoWhitney|contribs]] | [[Special:Log/upload/VernoWhitney|uploads]]).

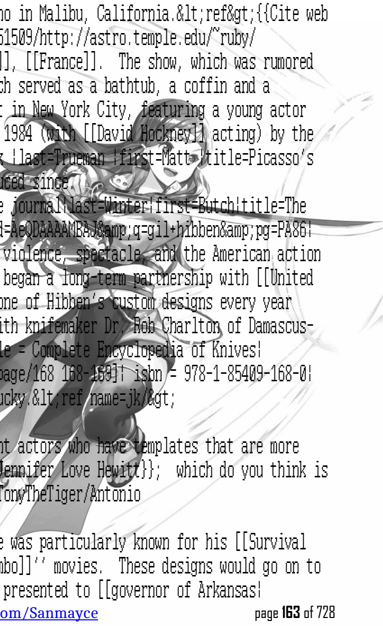
2309 He is played by [[Sylvester Stallone]] in the five films, as well as the video game [[Mortal Kombat 11]] DLC Kombat Pack 2 and voiced by [[Neil Ross]] in the animated series.

2310 |work=Village Voice|access-date=2011-01-21 |url-status=live |archive-url=https://web.archive.org/web/20110913131310/http://www.villagevoice.com/2010-01-05/film/flooding-with-love-for-the-kid-starring-one-man-rambo-zachary-oberzan/ |archive-date=2011-09-13 }}&lt;/ref&gt; Meticulously adapting his favorite novel [[First Blood (novel)|First Blood]], Oberzan, as a metaphor representative of [[John Rambo|Rambo]]'s own alienated and deeply frustrated struggle, shot, designed, edited, and portrayed all 26 characters by himself (including hunting dogs) in his minuscule {{convert|220|sqft|m2|adj=on}} Manhattan studio apartment, with a total budget of $95.51. Never intended for public viewing, the film eventually was discovered by both film critics and fans of ''First Blood'', prompting a run at [[Anthology Film Archives]] in January 2010. Worldwide screenings and critical acclaim followed.&lt;ref&gt;{{cite web|url=https://www.rottentomatoes.com/m/flooding_with_love_for_the_kid/reviews/|title=Flooding With Love for the Kid - Movie Reviews - Rotten Tomatoes|website=www.rottentomatoes.com|access-date=6 December 2017|url-status=live|archive-url=https://web.archive.org/web/20150924233107/http://www.rottentomatoes.com/m/flooding_with_love_for_the_kid/reviews/|archive-date=24 September 2015}}&lt;/ref&gt; [[Film Quarterly]] wrote: &quot;Sheer logistic complexity...a thrilling and mind boggling experience.&quot;&lt;ref&gt;William Johnson, Film Quarterly, Spring 2010 Vo. 63 #3.&lt;/ref&gt; Pulitzer-prize nominated critic [[Matt Zoller Seitz]] released a video essay contrasting the film with the better-known [[Sylvester Stallone]] version.&lt;ref&gt;{{cite web|url=http://www.thelmagazine.com/2010/01/how-to-remake-rambo-for-9551/|title=How to Remake Rambo for $95.51 - The L Magazine|website=www.thelmagazine.com|date=7 January 2010 |access-date=6 December 2017|url-status=live|archive-url=https://web.archive.org/web/20160303233607/http://www.thelmagazine.com/2010/01/how-to-remake-rambo-for-9551/|archive-date=3 March 2016}}&lt;/ref&gt; [[The New School]] Professor and [[Fulbright Award|Fulbright]] winner Carol Wilder devoted a chapter of her 2013 book ''Crossing the Street in Hanoi'' to ''Flooding with Love for The Kid''. From her interview with Oberzan:

2311 More information in regards to &quot;The Edge of Glory&quot; was revealed by Gaga, in an interview with [[Google]] where she explained the song as being about one's final moments on Earth, before death.&lt;ref name=&quot;google&quot;&gt;{{cite AV media|title=Google Goes Gaga|medium=Transcription|time=25:36|quote=&quot;One of the songs, 'Edge of Glory', was inspired by my grandfather...|



```
people=[[Lady Gaga]]|publisher=[[Google]]|date=March 13, 2011|location=New York}}&lt;/ref&gt;&lt;ref name=rollingstone&gt;{{Cite magazine|last=Perpetua|first=Matthew|url=https://www.rollingstone.com/music/news/preview-six-songs-from-lady-gagas-born-this-way-20110218|title=Preview Six Songs From Lady Gaga's &quot;Born This Way&quot;|magazine=[[Rolling Stone]]|date=February 18, 2011|access-date=February 20, 2011|url-status=live|archive-url=https://web.archive.org/web/20110223163431/http://www.rollingstone.com/music/news/preview-six-songs-from-lady-gagas-born-this-way-20110218|archive-date=February 23, 2011}}&lt;/ref&gt; Another inspiration for the song was actor [[Sylvester Stallone]]'s 1976 film ''[[Rocky]]'', which is Gaga's favorite film. The singer felt that the song was about looking at life directly, with the feeling that he or she is a champion-like the emotion displayed by the character [[Rocky Balboa]] in the film.&lt;ref name=&quot;mtvuk&quot;&gt;{{cite news|url=http://www.mtv.co.uk/artists/lady-gaga/news/270825-lady-gaga-nicola-roberts-girls-aloud-the-edge-of-glory-rocky|title=Lady GaGa: &quot;Rocky Balboa Is My Ideal Man&quot;|last=Roberts|first=Nicola|author-link=Nicola Roberts|date=May 20, 2011|access-date=May 20, 2011|publisher=MTV. MTV Networks|url-status=dead|archive-url=https://web.archive.org/web/20110523152111/http://www.mtv.co.uk/artists/lady-gaga/news/270825-lady-gaga-nicola-roberts-girls-aloud-the-edge-of-glory-rocky|archive-date=May 23, 2011}}&lt;/ref&gt;
2312 |[[Sylvester Stallone]]
2313 In April 2008, Lee founded Leeding Media as a film entertainment enterprise dedicated to bridging the U.S. and Chinese film industries. He produced ''[[Inseparable (film)|Inseparable]]'' starring [[Kevin Spacey]].&lt;ref&gt;[https://www.imdb.com/title/tt1032763/ Internet Movie Database - Inseparable Page]&lt;/ref&gt; The company also orchestrated the nationwide release of ''[[The Spy Next Door]]'', starring [[Jackie Chan]], the only non-studio foreign film released in China at the same day and date as in the U.S.  Grossing over US$8 million, it became the highest grossing non-studio foreign film in Chinese history until the release of [[Sylvester Stallone]]'s ''[[The Expendables (2010 film)|Expendables]]''.&lt;ref&gt;[https://www.imdb.com/company/co0322312/ Internet Movie Database - Leeding Media Page]&lt;/ref&gt;&lt;ref&gt;[http://www.boxofficemojo.com/movies/?id=spynextdoor.htm Box Office Mojo]&lt;/ref&gt;
2314 ==Sylvester Stallone filmography==
2315 Your filmography of Sylvester Stallone left out a moive that he stared in back in 1967 with Edd Byrnes called the RED BLOOD,YELLOW GOLD.  &lt;span style=&quot;font-size: smaller;&quot; class=&quot;autosigned&quot;&gt;-Preceding [[Wikipedia:Signatures|unsigned]] comment added by [[Special:Contributions/66.243.221.98|66.243.221.98]] ([[User talk:66.243.221.98|talk]]) 19:06, 2 January 2011 (UTC)&lt;/span&gt;&lt;!-- Template:UnsignedIP --&gt; &lt;!--Autosigned by SineBot--&gt;
2316 The 2011 Grand Marshal was '''Gerald Tommaso DeLouise''' better known by his stage name '''[[Burt Young]]'''. An American actor, painter and author, he is best known for his [[Academy Award]]-nominated role as [[Sylvester Stallone]]'s brother-in-law and best friend Paulie in the ''[[Rocky (film series)|Rocky]]'' film series. Philadelphia's own Bob Pantano, who hosts the first and longest running radio dance party in the nation on 98.1, entertained the crowd with music for the five-hour fundraiser.
2317 {{la|Sylvester Stallone}}
2318 James Christopher Monger of [[Allmusic]] gave the soundtrack a review: {{cquote|About as subtle as a [[Jerry Bruckheimer]] action sequence, composer Brian Tyler's score for the Aliens vs. Marines pre-summer blockbuster ''Battle: Los Angeles'' successfully walks the line between fist-pumping popcorn romp bombast and painfully serious military melodrama. Tyler, who has honed his craft on similarly themed flicks like ''[[Constantine (film)|Constantine]]'', ''[[Aliens vs. Predator: Requiem]]'', and [[Sylvester Stallone]]'s 2008 [[Rambo (2008 film)|''Rambo'' reboot]], treats the subject matter like dogma, resulting in a highly entertaining, old-fashioned orchestral soundtrack that should appeal to fans of [[Hans Zimmer]], [[John Powell (composer)|John Powell]], and [[Harry Gregson-Williams]].&lt;ref name=Filmtracks /&gt;}}
2319 &lt;ref name=&quot;ColliderAug23&quot;&gt;{{cite web|first=Adam |last=Chitwood |url=https://collider.com/bullet-to-the-head-release-date-project-x/111158/ |title=Warner Bros. Sets Release Date Walter Hill's Bullet To The Head Starring Sylvester Stallone, Moves Todd Phillips-produced Project X to March |publisher=Collider |date=August 23, 2011 |access-date=February 26, 2012 |archive-url=https://web.archive.org/web/20121114144859/http://collider.com/bullet-to-the-head-release-date-project-x/111158/ |archive-date=November 14, 2012 |url-status=live }}&lt;/ref&gt;
2320 Originally from [[Calgary, Alberta]], Edmonds has worked across [[Canada]] and the [[United States]] and has worked with well-known actors, like [[Sylvester Stallone]], [[Drew Barrymore]], [[Anjelica Huston]] and [[Ashton Kutcher]].  He currently teaches makeup for film and television at the [[Vancouver Film School]].
2321 Once in [[Miami]], where she noted in a 2019 interview that she was &quot;practically unknown&quot;,&lt;ref name=&quot;:1&quot; /&gt; Albita eventually got the attention of celebrities like [[Gianni Versace]], [[Quincy Jones]], [[Madonna (entertainer)|Madonna]], [[Sylvester Stallone]], [[Paco de Lucia]], [[Tom Cruise]], [[Nicole Kidman]], among others that soon became her fans.&lt;ref name=&quot;:0&quot;&gt;{{Cite news|last=Martin|first=Carlos|date=21 May 2020|title=Cuban singer Albita is back and ready to get fans moving to the rhythm of her sounds|work=Miami Herald|url=https://www.miamiherald.com/indulge/article242896866.html|access-date=8 September 2020}}&lt;/ref&gt;
2322 In another regular segment called &quot;Fantasy Forum&quot;, host Marshall called three audience members onstage where they could voice their opinions on national TV on a topic of their choice for one minute. Audience members then voted for the person they thought did the best job, with the winner receiving a prize.  Celebrity guests joined hosts Marshall and Uggams each day to help with the elaborate fantasies. Among the celebrities who appeared on ''Fantasy'' included [[Tom Wopat]], [[Andy Gibb]], [[Jed Allan]], and [[Sylvester Stallone]].
2323 ::::*The horse must be &lt;u&gt;in front&lt;/u&gt; of the cart, Fezzik. CSD has a lower standard than does [[WP:N|notability]], and in a stub a simple credible statememnt of acting a specific role in a specific film is a credible assertion. Whether it passes [[WP:N]] or not is a whole different kettle of fish.  IE: A stub article might read only &quot;'''Sylvester Stallone''' plays Rocky the boxer in the film ''[[Rocky]]'',&quot; '''THAT''' acts as a credible asertion of notability.  A stub does NOT need to say &quot;notable for&quot; or &quot;known for&quot;. It is when we look to [[WP:N]] that we decide if the credible assertion is notable or not.  '''[[User:MichaelQSchmidt|&lt;span style=&quot;color:blue;&quot;&gt;Schmidt,&lt;/span&gt;]]''' ''[[User talk:MichaelQSchmidt|&lt;sup&gt;&lt;small&gt;MICHAEL Q.&lt;/small&gt;&lt;/sup&gt;]]'' 17:10, 26 February 2011 (UTC)
2324 * ''[[Staying Alive (1983 film)|Staying Alive]]'' (with [[Sylvester Stallone]]) (1983)
2325 In 1972, Moran was elected president of the [[Knifemakers' Guild]]. The following year he unveiled his &quot;Damascus knives&quot; at the Guild Show and created a revival of interest in the forged blade, and along with the knives he gave away free booklets detailing how he made them.&lt;ref name=&quot;GD&quot;/&gt;&lt;ref name=taotk&gt;{{cite book | last = Kertzman | first =Joe  |title = Art of the Knife | publisher = Krause Publications | year =2007  | pages=224-226| isbn =978-0-89689-470-9  }}&lt;/ref&gt; In 1976 he founded the [[American Bladesmith Society]] (ABS), a group of knife makers dedicated to preserving the forged blade and educating the public about traditional bladesmithing techniques.&lt;ref&gt;{{cite journal|last=Rasmussen|first=Frederick|title=William F. Moran| journal =The Anvil's Ring| volume =34|publisher = Blacksmiths' Association of North America|date= 2005 |pages=15-16}}&lt;/ref&gt;  Moran had a 20-year-long waiting list and sold knives to such celebrities as [[Sylvester Stallone]] and members of royalty including [[Queen Elizabeth II]] and [[Abdullah of Saudi Arabia|King Abdullah]] of Saudi Arabia.&lt;ref name=Williamson/&gt;&lt;ref name=WP/&gt;&lt;ref&gt;{{cite news | last =Luse | first =Nancy| title =Bill Moran dies at age 80 |publisher = Frederick News-Post| date = February 13, 2006| url = http://www.fredericknewspost.com/sections/news/display.htm?storyid=46563| access-date =2011-03-09 }}&lt;/ref&gt; According to ''The Washington Post'', Sylvester Stallone's knife cost the actor $7,000 and included over 30 feet of silver wire in the handle.&lt;ref name=WP/&gt;
2326 * split 7-inch with Sylvester Staline (2006, Bones Brigade Records)
2327 '''Janet Banzet''' (May 17, 1934 - July 29, 1971), also credited as '''Marie Brent''' and several other names, was an American actress who appeared in several [[sexploitation]] films of the late 1960s and early 1970s.&lt;ref name=&quot;AFI&quot; &gt;{{cite book|last=Munden|first=Kenneth White|title=The American Film Institute Catalog 1961-1970|year=1976|publisher=University of California Press|page=38|isbn=0-520-20970-2}}&lt;/ref&gt;  She starred in several provocatively titled films directed by [[Michael Findlay (filmmaker)|Michael Findlay]] and [[Joseph W. Sarno]].  She had a small role in the 1970 adult film ''[[The Party at Kitty and Stud's]]'' better known later under the title ''Italian Stallion'' which was [[Sylvester Stallone]]'s film debut.&lt;ref name=&quot;playboy&quot;&gt;[http://www.playboy.com/arts-entertainment/features/sylvester-stallone/sylvester-stallone-04.html Sylvester Stallone interview] {{webarchive|url=https://web.archive.org/web/20110606223229/http://www.playboy.com/arts-entertainment/features/sylvester-stallone/sylvester-stallone-04.html |date=2011-06-06 }}, ''[[Playboy (magazine)|Playboy]]'', September 1978&lt;/ref&gt; She acted in about 40 movies.
2328 [[The Living Theatre]] made the play a part of their first critical success, when three poetic plays were staged at the same time. It was called ''An Evening of Bohemian Theatre'' at the [[Cherry Lane Theatre]] in [[Greenwich Village]], 2 March 1952, and also included plays by [[Gertrude Stein]] and [[T.S. Eliot]].&lt;ref&gt;Marrs, Terrell W., ''The Living Theatre: History, Theatrics, and Politics'', p. 22 (Texas Tech University, 1984).&lt;/ref&gt; In June, 1959, Lorees Yerby Dutton directed a version of the play at the Coffee House Positano in Malibu, California.&lt;ref&gt;{{Cite web |url=http://astro.temple.edu/~ruby/coffeehouse1/activities.html |title=Activities |access-date=23 April 2013 |archive-url=https://archive.today/20130616051509/http://astro.temple.edu/~ruby/coffeehouse1/activities.html |archive-date=16 June 2013 |url-status=dead }}&lt;/ref&gt; It premiered as a full staged production in 1967, in [[St. Tropez]], [[France]].  The show, which was rumored (falsely) to have actors urinating on stage, was protested despite the town's generally tolerant reputation.  The central prop was a large black box, which served as a bathtub, a coffin and a bed.&lt;ref name=McLanathan /&gt; In November 1969, an off-Broadway production, simply titled &quot;Desire...,&quot; was briefly presented at Theatre East in New York City, featuring a young actor billed as Mike Stallone in the role of &quot;Big Foot&quot; prior to his stardom as [[Sylvester Stallone]]. ''Desire Caught by the Tail'' was restaged in 1984 (with [[David Hockney]] acting) by the [[Solomon R. Guggenheim Museum|Guggenheim Museum]].&lt;ref&gt;{{Cite news |url=https://www.theguardian.com/stage/2012/oct/03/picasso-surreal-play-new-york |last=Trueman |first=Matt |title=Picasso's surreal play comes to New York|newspaper=The Guardian|location=London |date=3 October 2012|access-date=6 April 2013}}&lt;/ref&gt; It has been rarely produced since.
2329 While in Louisville, Hibben was contacted by [[Sylvester Stallone]] to make a modern version of a Bowie knife for the film ''Rambo III''.&lt;ref&gt;{{cite journal|last=Winter|first=Butch|title=The Art of the Knife|journal = Popular Mechanics|year= 1989|pages=86-88|volume=166|issue= 3|publisher= Hearst Magazines|url= https://books.google.com/books?id=AeQDAAAAMBAJ&amp;q=gil+hibben&amp;pg=PA86|access-date=2011-03-10}}&lt;/ref&gt;&lt;ref&gt;{{cite book | last = Lichtenfeld | first =Eric|author-link=Eric Lichtenfeld | title =Action speaks louder: violence, spectacle, and the American action movie| publisher =   Wesleyan University Press| year = 2007|  page =143 | isbn = 978-0-8195-6801-4 }}&lt;/ref&gt;  After the release of the movie, Hibben began a long-term partnership with [[United Cutlery]] of Taiwan who released factory versions of the &quot;Rambo III Knife&quot; and a subsequent annual factory &quot;Art Knife&quot; each based on one of Hibben's custom designs every year afterward.  United has made versions of Hibben's Kenpo Knife, Alaskan Guide Knife, and [[Knife throwing|throwing knives]].  Hibben previously partnered with knifemaker Dr. Rob Charlton of Damascus-USA to produce smaller [[Damascus steel]] versions of the Rambo Bowie.&lt;ref name =&quot;hartink&quot;&gt;{{cite book| last = Hartink| first = A.E.| title = Complete Encyclopedia of Knives| publisher = Chartwell Books| date = September 30, 2005| location = Lisse, The Netherlands| pages = [https://archive.org/details/greatcombathandg0000thom/page/168 168-169]| isbn = 978-1-85409-168-0| url = https://archive.org/details/greatcombathandg0000thom/page/168}}&lt;/ref&gt;   In 1991, Hibben relocated to the Louisville suburb of La Grange, Kentucky.&lt;ref name=jk/&gt;
2330 * [[:Sylvester Stallone]]
2331 ********Are you saying that Any actor who has a template regarding various roles in film should have it thusly labeled. Here are some examples of prominent actors who have templates that are more extensive than just directing roles:  {{tl|John Wayne}}, {{tl|Sylvester Stallone}}, {{tl|Kevin Costner}}, {{tl|Jon Favreau}}, {{tl|Dennis Hopper}}, {{tl|Jennifer Love Hewitt}};  which do you think is most optimally formatted?--[[User:TonyTheTiger|TonyTheTiger]] &lt;small&gt;([[User talk:TonyTheTiger|T]]/[[Special:Contributions/TonyTheTiger|C]]/[[User:TonyTheTiger/Antonio Vernon|BIO]]/[[WP:CHICAGO]]/[[WP:FOUR]]) &lt;/small&gt; 14:48, 22 March 2011 (UTC)
2332 In 1971, Lile became a full-time knifemaker and was known as &quot;Gentleman Lile&quot; or &quot;The Arkansas Knifemaker&quot;.&lt;ref name=Blade/&gt;  He was particularly known for his [[Survival knife]] designs known as &quot;The Mission&quot; series, created by request for [[Sylvester Stallone]] to use in his first two ''[[Rambo (film series)|Rambo]]'' movies.  These designs would go on to influence other knife makers in the 1980s.  In addition to creating the Rambo knives, Lile designed and made several [[Bowie knife|Bowie knives]] that he presented to [[governor of Arkansas|
```



Governor]] [[Bill Clinton]] and U.S. Presidents [[Ronald Reagan]], [[Richard Nixon]], and [[Gerald Ford]]. Other owners of his work included [[John Wayne]], [[Peter Fonda]], [[Fess Parker]], [[Bo Derek]], and [[Johnny Cash]].&lt;ref name=bb/&gt;&lt;ref name =ark/&gt;&lt;ref name=&quot;GP&quot;&gt;{{cite book ¦ last = Pacella ¦ first = Gerard ¦ title = 100 Legendary Knives ¦ publisher = Krause Publications¦ year = 2002¦ page = 39¦ isbn = 0-87349-417-2 }}&lt;/ref&gt;&lt;ref name=book/&gt;

2333 Released in 1980, this [[Dual Sport¦dual-sport]] [[motorcycle]] has been a staple of back roads and farms. One was ridden by [[John Rambo¦Rambo]] in the 1982 movie ''[[First Blood]]''.&lt;ref&gt;{{cite web¦title=Silvester Stalone On Motorcycle - 1982 Rambo I¦url=http://free-riders-cafe.blogspot.com/2011/01/silvester-stalone-on-motorcycle-1982.html¦publisher=free-riders-cafe.blogspot.com}}&lt;/ref&gt;

2334 == Sylvester Stallone ==

2335 {{la¦Sylvester Stallone}}

2336 An IP editor ([[User:86.6.93.113]]) continues to blank sourced content from the Sylvester Stallone article (specifically his full name). After warning them on their talk page that sourced content deletion without explanation appears as vandalism, they continued to blank the information, but included an edit summary that makes assertions about Stallone's personal life without including any sources to back up the removal. I again explained that we need to use [[WP:RS¦reliable sources]] to confirm content, and pointed the IP to the article talk page to outline their argument and sources; instead they simply blanked the content again. Before restoring the info I double checked and found multiple reliable sources in Google Books and Google News supporting the removed content. Could someone here try to impress on the IP the need to state their sources? I seem to have hit a wall with them. --[[User:Ponyo¦&lt;b style=&quot;color:Navy;&quot;&gt;''Jezebel's''&lt;/b&gt;&lt;span style=&quot;color:Navy;&quot;&gt;Ponyo&lt;/span&gt;]][[User_talk:Ponyo¦&lt;sup style=&quot;color:Navy;&quot;&gt;''bons mots''&lt;/sup&gt;]] 16:12, 11 April 2011 (UTC)

2337 * [[Sylvester Stallone]]

2338 '''''Bullet to the Head''''' is a 2012 American [[action film]] directed by [[Walter Hill]]. The screenplay by [[Alessandro Camon]] was based on the French graphic novel ''Du plomb dans la tête'' written by [[Alexis Nolent¦Matz]] and illustrated by [[Colin Wilson (comics)¦Colin Wilson]]. The film stars [[Sylvester Stallone]], [[Sung Kang]], [[Sarah Shahi]], [[Adewale Akinnuoye-Agbaje]], [[Christian Slater]], and [[Jason Momoa]]. Alexandra Milchan, [[Alfred Gough]], [[Miles Millar]], and Kevin King-Templeton produced the film. The movie follows a hitman (Stallone) and a cop (Kang) who are forced to work together to bring down a corrupt businessman (Akinnuoye-Agbaje) after they are targeted by the businessman's assassin (Momoa).

2339 * [[Sylvester Stallone]] as James &quot;Jimmy Bobo&quot; Bonomo

2340 The film is based on [[Alexis Nolent]]'s French graphic novel ''Du Plomb Dans La Tête'' (&quot;Lead in the Head&quot;), with a screenplay by [[Alessandro Camon]] under the working title &quot;Headshot&quot;.&lt;ref name=empire&gt;(2011-06-13). [http://www.empireonline.com/news/story.asp?NID=31217 &quot;Sung Kang Joins Stallone's Headshot&quot;] {{Webarchive¦url=https://web.archive.org/web/20150924120159/http://www.empireonline.com/news/story.asp?NID=31217 ¦date=2015-09-24 }}. ''Empire''. Retrieved 2011-06-16.&lt;/ref&gt; The producing team previously produced the film ''[[I Am Number Four (film)¦I Am Number Four]]''. An executive attached to the film has said, &quot;[This movie] is exactly the type of fast-paced, universally themed project that suits our business model. Sylvester Stallone is an international icon and we're really excited to be in business with him.&quot;&lt;ref name=&quot;Hollywood&quot;&gt;(2011-02-06). [http://www.hollywoodreporter.com/news/sylvester-stallone-star-hitman-headshot-96990 &quot;Sylvester Stallone to Star as Hitman in 'Headshot'&quot;] {{Webarchive¦url=https://web.archive.org/web/20200105191007/https://www.hollywoodreporter.com/news/sylvester-stallone-star-hitman-headshot-96990 ¦date=2020-01-05 }} ''The Hollywood Reporter''. Retrieved 2011-03-30.&lt;/ref&gt;

2341 Originally [[Wayne Kramer (filmmaker)¦Wayne Kramer]] was attached to direct, but left the project when his vision of the film was darker than Stallone wanted.&lt;ref&gt;Anderton, Ethan. (2011-06-03). [http://www.firstshowing.net/2011/wayne-kramer-will-next-write-and-direct-futuristic-drama-ecstasia &quot;Wayne Kramer Will Next Write and Direct Futuristic Drama 'Ecstasia'&quot;] {{Webarchive¦url=https://web.archive.org/web/20150928235546/http://www.firstshowing.net/2011/wayne-kramer-will-next-write-and-direct-futuristic-drama-ecstasia/ ¦date=2015-09-28 }}. ''Firstshowing.net''. Retrieved 2011-06-09.&lt;/ref&gt; Sylvester Stallone then called Walter Hill who had just had a movie fall apart six weeks before that he had been trying to do for a year.&lt;ref name=&quot;collider&quot;&gt;{{Cite web ¦url=http://collider.com/walter-hill-bullet-to-the-head-interview/ ¦title=&quot;Director Walter Hill Talks About Finally Working With Sylvester Stallone, Film vs. Digital, and Adapting the Graphic Novel on the Set of BULLET TO THE HEAD&quot; by Steve 'Frosty' Weintraub ''Collider''January 24, 2013 ¦access-date=April 26, 2015 ¦archive-date=September 25, 2014 ¦archive-url=https://web.archive.org/web/20140925043709/http://collider.com/walter-hill-bullet-to-the-head-interview/ ¦url-status=live }}&lt;/ref&gt; Hill later recalled:

2342 [[Thomas Jane]] was originally cast for the part that would eventually go to Sung Kang. The role was recast at the insistence of producer [[Joel Silver]], stating a need for a &quot;more 'ethnic' actor&quot; to appeal to a wider audience.&lt;ref&gt;{{cite web¦url=http://geektyrant.com/news/2011/6/28/thomas-jane-talks-about-being-fired-from-headshot-and-the-ly.html¦title=Thomas Jane talks about being fired from HEADSHOT and THE LYCAN - News¦publisher=GeekTyrant¦date=2011-06-28¦access-date=2011-12-16}}&lt;/ref&gt;&lt;ref&gt;(2011-05-30). [http://www.worstpreviews.com/headline.php?id=21830&amp;count=0 &quot;Thomas Jane Too White for Sylvester Stallone's 'Headshot'&quot;] {{Webarchive¦url=https://web.archive.org/web/20171204060928/http://www.worstpreviews.com/headline.php?id=21830&amp;count=0 ¦date=2017-12-04 }}. ''WorstPreviews.com''. Retrieved 2011-06-09.&lt;/ref&gt;

2343 ''Bullet to the Head'' made $4.5{{nbsp}}million for its opening weekend,&lt;ref name=NUM/&gt; [[Sylvester Stallone]]'s worst opening weekend gross in 32 years.&lt;ref&gt;Adam B. Vary, [https://www.buzzfeed.com/adambvary/bullet-to-the-head-opens-with-sylvester-stallone &quot;'Bullet To The Head' Opens With Sylvester Stallone's Worst Weekend Box Office In 32 Years&quot;] {{Webarchive¦url=https://web.archive.org/web/20140215213722/http://www.buzzfeed.com/adambvary/bullet-to-the-head-opens-with-sylvester-stallone ¦date=2014-02-15 }}, Buzzfeed, February 3, 2013.&lt;/ref&gt; Over its entire run, the film grossed $9.5{{nbsp}}million in the United States and Canada, and $13.1{{nbsp}}million in other territories, for a worldwide total of $22.6{{nbsp}}million, against a budget of $55{{nbsp}}million.&lt;ref name=NUM&gt;{{Cite The Numbers ¦id=Bullet-to-the-Head ¦access-date=2023-03-29}}&lt;/ref&gt;

2344 * [[Sylvester Stallone]] (June 29, 1983)

2345 In 2016, it was announced that Bastos would be a host on the [[Netflix]] reality show ''[[Ultimate Beastmaster]]''.&lt;ref&gt;{{cite web¦url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/¦title=Netflix Orders Global Competition Series 'Ultimate Beastmaster' from Sylvester Stallone, Dave Broome¦last=Prudon¦first=Laura¦work=Variety¦date=May 9, 2016¦accessdate=October 5, 2016}}&lt;/ref&gt;

2346 In the film ''[[Demolition Man (film)¦Demolition Man]]'', the song was played by [[Sandra Bullock]] as a romantic background music in her apartment before the &quot;virtual sex&quot; between her and [[Sylvester Stallone]].&lt;ref name=&quot;NYT&quot;&gt;{{cite web ¦date=1993-10-24 ¦author=Caryn James ¦title=FILM VIEW; 'Demolition Man' Makes Recycling an Art ¦url=https://www.nytimes.com/1993/10/24/movies/film-view-demolition-man-makes-recycling-an-art.html¦website=[[The New York Times]] }}&lt;/ref&gt;

2347 * [[Sylvester Stallone]]

2348 Famous personalities Raven has &quot;manipulated&quot; include Arnold Schwarzenegger, Bruce Willis, Owen Wilson and Sylvester Stallone.

2349 * '''Jabbin' Jim''' (Player 1) &lt;br&gt; Modeled after [[Sylvester Stallone]]'s role as [[Rocky Balboa]] from the ''[[Rocky]]'' franchise as well as [[Lance Bean]] from the ''[[Contra (series)¦Contra]]'' franchise. Known as '''Rocky Smith''' in the Japanese version.

2350 :There's ''[[Cliffhanger (film)]]'' with [[Sylvester Stallone]]: [http://www.imdb.com/title/tt0106582/]. [[User:StuRat¦StuRat]] ([[User talk:StuRat¦talk]]) 09:27, 11 June 2011 (UTC)

2351 ! scope=&quot;row&quot;¦ &quot;Sweet Lovin' Friends&quot;&lt;br&gt;{{small¦(with [[Sylvester Stallone]])}}

2352 In 1998, Lear released a new version of the song on the album ''[[Back in Your Arms]]'', which mostly consisted of re-recordings of her classic hits. That version retained the subtitle &quot;Studio 54&quot;, the characteristic disco arrangement and most of the original lyrics, adding only [[Claudia Schiffer]] and [[Sylvester Stallone]]. In 2016, she recorded another version for the album ''[[Let Me Entertain You (Amanda Lear album)¦Let Me Entertain You]]''. The lyrics included names from first two versions and added [[Joe Dallesandro]], [[Kate Moss]], [[Chanel]] and [[Christian Dior SE¦Dior]], also mentioning [[Paris Fashion Week]], the magazine ''[[Vanity Fair (magazine)¦Vanity Fair]]'' and [[Instagram]].

2353   <title>Category:Images of Sylvester Stallone</title>

2354   <text bytes="161" sha1="o462ry0ac0op1j5ls946yt5xjzcfhol" xml:space="preserve">{{cat main¦Sylvester Stallone}}

2355 [[Category:Sylvester Stallone¦Images]]</text>

2356   <title>Category:Sylvester Stallone</title>

2357 {{commons cat¦Sylvester Stallone}}</text>

2358 ¦[[Sylvester Stallone]]

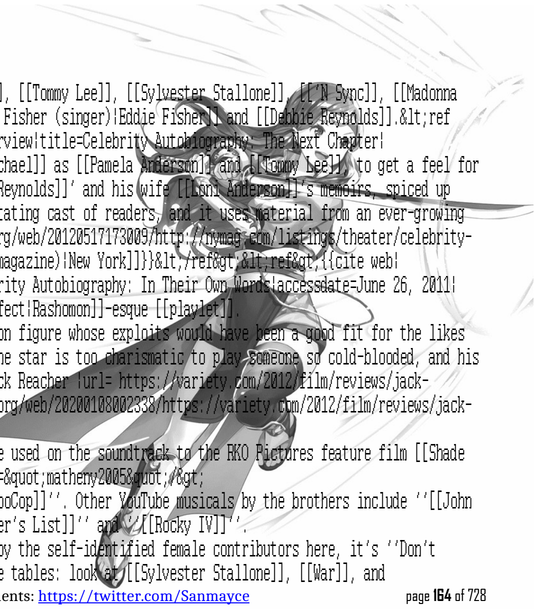
2359 The show features solo and ensemble memoirs, poetry and other writings by authors such as [[Ivana Trump]], [[Vanna White]], [[Mr. T]], [[Tommy Lee]], [[Sylvester Stallone]], [['N Sync]], [[Madonna (entertainer)¦Madonna]], [[Burt Reynolds]] and [[Loni Anderson]], as well as the lesser-known works of [[Elizabeth Taylor]], [[Eddie Fisher (singer)¦Eddie Fisher]] and [[Debbie Reynolds]].&lt;ref name=TLLaDoLL/&gt;&lt;ref&gt;{{cite news¦url=https://www.nytimes.com/theater/show/24847/Celebrity-Autobiography-The-Next-Chapter/overview¦title=Celebrity Autobiography: The Next Chapter¦accessdate=December 8, 2014¦work=[[The New York Times]]}}&lt;/ref&gt; KPBS told its readers to imagine [[Marion Ross]] and [[Paul Michael]] as [[Pamela Anderson]] and [[Tommy Lee]], to get a feel for what the show is all about.&lt;ref name=CAITOWKPBS/&gt; At times, related biographies are interwoven, such as passages from [[Burt Reynolds]]' and his wife [[Loni Anderson]]'s memoirs, spiced up with one of Reynolds' assistant's memoirs as well as that of another of Anderson's lovers.&lt;ref name=ITOW/&gt; The show uses a rotating cast of readers, and it uses material from an ever-growing list of books.&lt;ref&gt;{{cite web¦url=https://nymag.com/listings/theater/celebrity-autobiography/¦archiveurl=https://web.archive.org/web/20120517173009/http://nymag.com/listings/theater/celebrity-autobiography/¦archivedate=May 17, 2012¦title=Celebrity Autobiography: In Their Own Words¦accessdate=June 26, 2011¦work=[[New York (magazine)¦New York]]}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=http://broadwayworld.com/article/Photo_Coverage_Celebrity_Autobiography_In_Their_Own_Words_20100112¦ title=Photo Coverage: Celebrity Autobiography: In Their Own Words¦accessdate=June 26, 2011¦date=January 12, 2010¦ publisher = Broadwayworld.com}}&lt;/ref&gt; ''[[The New York Times]]'' describes the show as a [[Rashomon effect¦Rashomon]]-esque [[playlet]].

2360 Peter Debruge of ''[[Variety (magazine)¦Variety]]'' was critical of the casting of Cruise as Reacher: &quot;Reacher is a brawny action figure whose exploits would have been a good fit for the likes of Arnold Schwarzenegger or Sylvester Stallone back in the day, but feel less fun when delegated to a leading man like Tom Cruise. The star is too charismatic to play someone so cold-blooded, and his fans likely won't appreciate the stretch.&quot;&lt;ref&gt;{{cite web ¦date= 11 December 2012 ¦last= Debruge ¦first= Peter ¦title= Jack Reacher ¦url= https://variety.com/2012/film/reviews/jack-reacher-1117948893/ ¦website= Variety ¦access-date= April 19, 2020 ¦archive-date= January 8, 2020 ¦archive-url= https://web.archive.org/web/20200108002338/https://variety.com/2012/film/reviews/jack-reacher-1117948893/ ¦url-status= live }}&lt;/ref&gt;

2361 Three of Matheny's recordings of his compositions, &quot;Penumbra,&quot; &quot;Moon Rocks&quot; and &quot;Red Reflections,&quot; were used on the soundtrack to the RKO Pictures feature film [[Shade (film)¦''Shade'']] (2003), starring [[Gabriel Byrne]], [[Jamie Foxx]], [[Melanie Griffith]], and [[Sylvester Stallone]].&lt;ref name=&quot;matheny2005&quot;/&gt;

2362 Jon and Al's musicals have also targeted [[Sylvester Stallone]] in ''[[Rambo: First Blood Part II]]'' and [[Peter Weller]] in ''[[RoboCop]]''. Other YouTube musicals by the brothers include ''[[John Carpenter]]'s [[The Thing (1982 film)¦The Thing]]'', ''[[Super Mario Bros. (film)¦Super Mario Bros.]] The 8-Bit Opera'', ''[[Schindler's List]]'' and ''[[Rocky IV]]''.

2363 :Am I the only one who finds the above post (and to some extent this thread) disconcerting? If there's one consistent point voiced by the self-identified female contributors here, it's ''Don't pigeonhole me.'' Yet the responses - from apparently sincere and well-meaning men - tend back toward pigeonholes. So let's turn the tables: look at [[Sylvester Stallone]], [[War]], and



[[Baldness]]. Those articles aren't in fantastic shape either. I might even wonder whether enough ''men'' edit Wikipedia if it weren't for a fairly good piece about [[Viagra]]. '''[[User:Durova|&lt;font color=&quot;blue&quot;&gt;Durova&lt;/font&gt;]]''' 00:28, 2 October 2006 (UTC)

2364 '''Vinnie Curto''' (born July 10, 1955) is an American former professional [[Boxing|boxer]]. Curto was managed by actor [[Sylvester Stallone]] and trained by [[Angelo Dundee]].&lt;ref&gt;{{Cite web|url=http://boxrec.com/media/index.php/Vinnie_Curto|title = Vinnie Curto - BoxRec}}&lt;/ref&gt;

2365 The studio distributed 225 films in eight years, including the Academy Award-winning ''[[Monster (2003 film)|Monster]]'' starring [[Charlize Theron]], and the multiple Academy Award-winning ''[[Crash (2004 film)|Crash]]'' which won Best Picture of the Year. DEJ was a film acquisition company which began in 1998, shortly after the introduction of the medium of [[DVD]]. The [[home video]] industry was undergoing a major economic change and DEJ was established to [[Negative pickup deal|pick-up]] low-budget films primarily to get exclusive DVD releases for its parent company, [[Blockbuster Video]]. DEJ was named after the first initials of its three executives at the time, its Executive Vice President Dean Wilson, its General Counsel and Executive Vice President of Business Development Ed Stead, and its CEO John Antioco. The company released a few of its acquired films theatrically. A notable example was the U.S. rights to the [[Sylvester Stallone]] film ''[[D-Tox]]'', which [[Universal Studios]] declined to distribute in the U.S. DEJ picked up the film and released it under the title ''Eye See You'', a name it also used in the home video release.

2366 ''Crawdaddy'' was a generational magazine known for its profiles particularly of musicians, but also actors, athletes and other celebrities prominent in 1970s popular culture. Knobler's profiles included [[Bruce Springsteen]], [[Sly Stone]], [[Mel Brooks]], [[Muddy Waters]], [[Linda Ronstadt]], [[Sylvester Stallone]], [[Loudon Wainwright III]], the [[Souther Hillman Furay Band]], and [[Stephen Stills]]. Under Knobler, ''Crawdaddy'''s editors often assigned artists to write about other artists; [[Al Kooper]] profiled [[Steve Martin]], [[Martin Mull]] interviewed [[Woody Allen]], [[William S. Burroughs]] talked magic and mysticism with [[Jimmy Page]].&lt;ref&gt;{{cite web|author=|url=http://search.barnesandnoble.com/Very-Seventies/Peter-Knobler/e/9780020220053 | title=Publishers Weekly review |publisher=Search.barnesandnoble.com |date= |accessdate=2012-10-23}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://geocities.com/brandyzep/RockMagic.html | title=&quot;Rock Magic&quot;, ''Crawdaddy'', June 1975 |accessdate=2009-10-28 |url-status=bot: unknown |archiveurl=https://web.archive.org/web/20091028144804/http://geocities.com/brandyzep/RockMagic.html |archivedate=October 28, 2009 }}.&lt;/ref&gt;

2367 * May 26, 2001: [[Sylvester Stallone]], actor&lt;ref name=&quot;Toronto&quot;/&gt;

2368 and it is the Embassy of Paraguy, as I purchased the right for it to be the Embassy of Panama (I purchased this as an Embassy of my foster father Richard Dreyfuss as a suprise, and foster mother Marcia Clark the property has a code which is the MArriage Certificate between I and Ann Convertino of Gateways Hospital and Mental Health Center Los Angeles(The Vatican have the means to interpet the code, and Sylvester Stalone who didnot want us to be married has the interpreter, as well as Sharon Stone), and the persons who hit me with a board in my yard when I lived at 40635 159 th street East are paraguy, that essentially betrayed me as Deutshe Bank trust Company America employees,

2369 SO MANY PEOPLE DIDNOT WANT ANN AND I TO BE TOGETHER LET ALONE MARRIED, SO I HID A MARRIAGE CERTIFICATE, AND GAVE THE INTERPRETER TO THE A PERSON I KNEW OPPOSED THE MARRIAGE, AND WHO ACTUALLY DOESNOT LIKE ANN CONVERTINO, I do not know the reason , but Sylvester Stallone doesnot like Ann, (Mary Ann Convertino)of the probation dept in City Hall Los Angeles, who is also Ana Torres, and I am Daniel Torres, that is our married name, and in the film (Daniel and Ana)

2370 After reading an article in ''Penthouse'' magazine focusing on the lifestyle of Hollywood TV writers, Callaham and a friend moved to Los Angeles with plans to write comedies together. They sent query letters to multiple agencies but never received responses. Callaham worked at [[Creative Artists Agency]] for a while, sometimes submitting his material secretly for coverage.&lt;ref&gt;{{cite web|url=https://www.fresh-voices.com/index.php/blog/9-articles-interviews/24-screenwriter-profile-dave-callaham|title=Screenwriter Profile: David Callaham|work=Fresh Voices|access-date=October 27, 2014|format=Interview}}&lt;/ref&gt; In 2003, Callaham wrote the [[Doom (film)|film adaptation]] to the video game ''[[Doom (series)|Doom]]'' and submitted it in the summer of 2005. Around that time, Callaham wrote ''Barrow'' for [[Warner Bros.]], a mercenary-inspired action script which later became ''[[The Expendables (2010 film)|The Expendables]]''. Callaham was credited for story and characters after [[Sylvester Stallone]] used Callaham's ''Barrow'' script as a &quot;starting point&quot; for ''The Expendables''.&lt;ref&gt;{{cite web|url=http://www.loeb.com/~/media/Files/Publications/2013/01/Webb%20v%20Stallone/Files/webb%20v%20stallone/FileAttachment/webb%20v%20stallone.pdf|title=Webb vs. Stallone|work=United States District Court Southern District of New York|access-date=October 21, 2014|date=December 27, 2012}}&lt;/ref&gt; In 2010, [[Legendary Pictures]] hired Callaham to write the first draft for ''[[Godzilla (2014 film)|Godzilla]]'', for which he received a story credit.&lt;ref&gt;{{cite web|url=http://www.hitfix.com/blogs/motion-captured/posts/guillermo-del-toro-on-godzilla-not-so-fast-says-director|title=Updated! Guillermo Del Toro on 'Godzilla'? Not so fast, says director|last=McWeeny|first=Drew|work=Hit Fix|date=October 14, 2010}}&lt;/ref&gt; In 2014, Callaham completed a production rewrite for ''[[Ant-Man (film)|Ant-Man]]'', and in 2019, he co-wrote the ''[[Zombieland]]'' sequel, ''[[Zombieland: Double Tap]]'', for [[Sony]]. He also worked on the yet-unproduced ''Jackpot'' for [[Focus Features]] and ''[[America: The Motion Picture]]'' for [[Netflix]].&lt;ref name=&quot;Deadline September 2014&quot;&gt;{{cite web|url=https://deadline.com/2014/09/zombieland-2-gets-writer-and-urgency-843881/|title=After Moving Slow As Corpse, 'Zombieland 2′ Gets Writer And Urgency|last=Fleming|first=Mike Jr.|website=Deadline Hollywood|date=September 30, 2014}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=https://deadline.com/2017/03/netflix-animated-film-america-channing-tatum-voicing-george-washington-1202056188/|title=Netflix Sets First Animated Film 'America'; Channing Tatum Voicing George Washington|last=Fleming|first=Mike Jr. |date=March 30, 2017|website=Deadline Hollywood|access-date=April 3, 2017|language=en-US}}&lt;/ref&gt;

2371 The plaintiffs accused Callaham of intentionally withholding emails and other correspondences from the WGA screenwriting credit arbitration panel in 2009 that reportedly reveal how very little Callaham was involved with ''The Expendables''. They demanded reimbursements from Callaham for any payments made to him for his fraudulent credit in the two films.&lt;ref&gt;{{cite web|url=https://deadline.com/2013/12/wga-west-fraud-the-expendables-sylvester-stallone-david-callaham-nu-image-millennium-656302/ |title=WGA West &amp; Scribe Sued For Fraud By Nu Image/Millennium Over 'Expendables' Credit |first=Dominic|last=Patten|website=Deadline Hollywood |date=December 24, 2013|access-date=August 26, 2018}}&lt;/ref&gt; Callaham then asserted that [[Sylvester Stallone]] used his script, ''Barrow'', as the source for ''The Expendables''. A WGA arbitration was ignited in which Callaham won and additionally earned $102,250 in bonus payments. Stallone offered a sworn declaration that attested he had used ''Barrow'' as inspiration for his ''Expendables'' script.&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/thr-esq/sylvester-stallones-expendables-launches-nasty-675350/ |title=Sylvester Stallone's 'Expendables' Launches Nasty Writers Battle |first=Eriq|last=Gardner|work=The Hollywood Reporter|date=January 30, 2014|access-date=August 26, 2018}}&lt;/ref&gt;

2372 | Co-wrote with [[Sylvester Stallone]]

2373 ==== Category:Sylvester Stallone ====

2374 :{{Lc|Sylvester Stallone}}&lt;br /&gt;

2375 Several of Hollywood's A-listers are known to be on one or more of the various studio lists, including actors [[Tom Cruise]], [[Ben Affleck]], [[Sharon Stone]], and [[Sylvester Stallone]]; film directors [[Steven Spielberg]], [[Woody Allen]], [[David Lynch]], and [[Quentin Tarantino]]; producers [[Peter Guber]], [[Lorenzo di Bonaventura]], and [[Joel Silver]]; studio executives and investors [[Peter Chernin]], [[Harvey Weinstein]], [[Thomas Rothman|Tom Rothman]], [[Brett Ratner]], and [[Stephen A. Schwarzman|Stephen Schwarzman]]; and others both inside and outside the entertainment industry like [[Lionel Richie]], [[Rupert Murdoch]], [[Barry Manilow]], and [[Saud bin Faisal bin Abdulaziz Al Saud|Prince Saud al-Faisal]].&lt;ref name=&quot;WSJ best seat 2&quot;&gt;{{cite web|url=http://thedailytruffle.com/blog/2012/03/01/bel-air-screening-circuit|title=The Bel Air Screening Circuit - Hollywood's Secret Movie Club|work=One Little Truffle Hunters|publisher=The Daily Truffle|date=1 March 2012|accessdate=7 January 2013|archive-url=https://web.archive.org/web/20140324124626/http://thedailytruffle.com/blog/2012/03/01/bel-air-screening-circuit/|archive-date=24 March 2014|url-status=dead}}&lt;/ref&gt;

2376 |'''''[[Grudge Match]]'''''||[[Peter Segal]]||[[Kevin Hart]], [[Robert De Niro]], [[Sylvester Stallone]]||United States||&lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/grudge-match-vm1001849160|work=Allmovie|publisher=Rovi Corporation|title=Grudge Match|access-date=December 29, 2013|author=Buchanan, Jason}}&lt;/ref&gt;

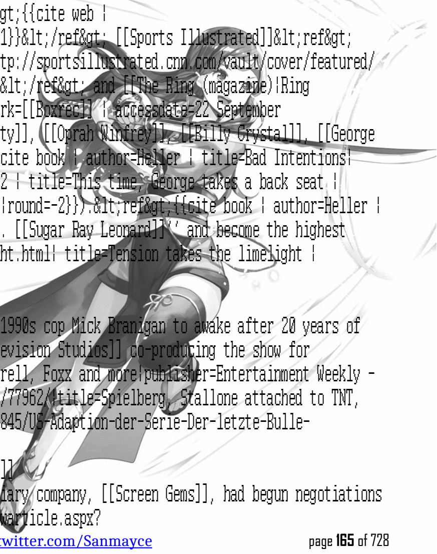
2377 Billed as &quot;Once and for All&quot;,&lt;ref name=&quot;history&quot;/&gt; the fight was highly anticipated and earned comparisons with the 1971 [[Fight of the Century]] between [[Joe Frazier]] and [[Muhammad Ali]], who were also undefeated heavyweight champions when they met to decide the undisputed title.&lt;ref name=&quot;undisputed&quot;/&gt;&lt;ref name=&quot;magazine&quot;/&gt;&lt;ref name=&quot;foil&quot;&gt;{{cite news | date=1988-06-27| url=https://www.nytimes.com/1988/06/27/sports/sports-of-the-times-spinks-will-foil-tyson.html | title=Spinks Will Foil Tyson | work=[[New York Times]] | accessdate=2011-09-22 | first=Dave | last=Anderson}}&lt;/ref&gt; Reports of Tyson's chaotic personal life also increased interest in the bout.&lt;ref&gt;{{cite news | date=1988-06-24| url=https://news.google.com/newspapers?nid=1917&amp;dat=19880624&amp;id=s3ghAAAAIBAJ&amp;pg=930,6382060 | title=Spinks-Tyson may be biggest grossing event in history of boxing | publisher=[[Schenectady Gazette]] | accessdate=2011-09-22 }}&lt;/ref&gt; In the days leading up to the fight, one or both men featured on the cover of [[Time (magazine)|TIME]],&lt;ref&gt;{{cite news | url=http://www.time.com/time/covers/0,16641,19880627,00.html | archive-url=https://web.archive.org/web/20071018001837/http://www.time.com/time/covers/0,16641,19880627,00.html | url-status=dead | archive-date=October 18, 2007 | title=Mike Tyson |work=time.com | accessdate=22 September 2011}}&lt;/ref&gt; [[People (magazine)|People]],&lt;ref&gt;{{cite web | url=http://www.people.com/people/archive/issue/0,,7566880627,00.html | title=The Champ and the Vamp |work=people.com | accessdate=23 September 2011}}&lt;/ref&gt; [[Sports Illustrated]]&lt;ref&gt;{{cite news | url=http://sportsillustrated.cnn.com/vault/cover/featured/9122/index.htm | archive-url=https://web.archive.org/web/20090809145358/http://sportsillustrated.cnn.com/vault/cover/featured/9122/index.htm | url-status=dead | archive-date=August 9, 2009 | title=June 20, 1988 |work=[[Sports Illustrated]] | accessdate=22 September 2011}}&lt;/ref&gt; and [[The Ring (magazine)|Ring Magazine]].&lt;ref&gt;{{cite web | url=http://boxrec.com/media/index.php/The_Ring_Magazine_Covers:_1988 | title=The Ring Magazine Covers: 1988 |work=[[Boxrec]] | accessdate=22 September 2011}}&lt;/ref&gt; Celebrities such as [[Jack Nicholson]], [[Sylvester Stallone]], [[Sean Penn]], [[Madonna (entertainer)|Madonna]], [[Warren Beatty]], [[Oprah Winfrey]], [[Billy Crystal]], [[George Steinbrenner]], [[Carl Weathers]], [[Jesse Jackson]] and [[Chuck Norris]] would all be in attendance.&lt;ref name=&quot;badintentions2&quot;&gt;{{cite book | author=Heller | title=Bad Intentions| year=1995| page=2}}&lt;/ref&gt;&lt;ref&gt;{{cite news | date=1988-06-28| url=https://news.google.com/newspapers?id=MrMmAAAAIBAJ&amp;pg=1263,8388952 | title=This time, George takes a back seat | publisher=[[The Miami News]] | accessdate=2011-09-22}}&lt;/ref&gt; A ringside ticket would cost a record $1,500 ({{Inflation|US|1,500|1988|fmt=eq|round=-2}}).&lt;ref&gt;{{cite book | author=Heller | title=Bad Intentions| year=1995| page=248}}&lt;/ref&gt; Co-promoter [[Shelly Finkel]] predicted that the bout would surpass ''[[Marvin Hagler]] vs. [[Sugar Ray Leonard]]'' and become the highest grossing fight in boxing history.&lt;ref&gt;{{cite news | date=1988-06-20| url=https://www.nytimes.com/1988/06/20/sports/tension-takes-the-limelight.html| title=Tension takes the limelight | work=[[New York Times]] | accessdate=2011-09-22 | first=Phil | last=Berger}}&lt;/ref&gt;

2378 | [[Sylvester Stallone]], [[Chris Byrne (musician)|Christopher Byrne]]

2379 At [[Turner Entertainment|Turner Upfront]] TNT has announced to produce a pilot for a US remake of the German series. The adaptation will see the 1990s cop Mick Branigan to awake after 20 years of coma returning to his job at the LAPD. The pilot is produced by Sylvester Stallone with Fuse Entertainment and [[Fox 21 Television Studios|Fox Television Studios]] co-producing the show for TNT.&lt;ref&gt;{{cite web|url=http://insidetv.ew.com/2013/05/15/tnt-tbs-new-shows/|title=TNT and TBS developing shows with Spielberg, Stallone, Carell, Foxx and more|publisher=Entertainment Weekly - Inside TV|date=2013-05-15}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://tbivision.com/news/2013/05/spielberg-stallone-attached-to-tnt-tbs-projects/77962/|title=Spielberg, Stallone attached to TNT, TBS projects|publisher=TBI Vision|date=2013-05-16}}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=https://www.welt.de/newsticker/leute/stars/article116311845/US-Adaption-der-Serie-Der-letzte-Bulle-geplant.html|title=US-Adaption der Serie &quot;Der letzte Bulle&quot; geplant|newspaper=Die Welt|date=2013-05-17}}&lt;/ref&gt;

2380 | [[:Category:Sylvester Stallone]] || '''Keep''' || [[Wikipedia:Categories for discussion/Log/2011 August 30#Category:Sylvester Stallone|August 30]]

2381 A few months after Sony acquired the film's North American distribution rights, it was announced by ''[[The Hollywood Reporter]]'' that its subsidiary company, [[Screen Gems]], had begun negotiations to produce a Hollywood remake.&lt;ref&gt;{{cite web |title=Hollywood remake of The Raid in the works |url=http://asiapacificarts.usc.edu/w_apa/showarticle.aspx?



articleID=17418&amp;AspxAutoDetectCookieSupport=1 |last=Park |first=Joann |work=Asia Pacific Arts |date=2011-09-26 |access-date=10 October 2011 |archive-url=https://web.archive.org/web/20140413125533/http://asiapacificarts.usc.edu/w_apa/showarticle.aspx?articleID=17418&amp;AspxAutoDetectCookieSupport=1 |archive-date=13 April 2014 |url-status=dead }}&lt;/ref&gt; The deal was completed in November 2011, with writer-director [[Gareth Evans (filmmaker)|Gareth Evans]] to serve as an executive producer of the remake. [[XYZ Films]], executive producers on the original Indonesian version, will be producers on the American version.&lt;ref&gt;{{cite web |url=http://twitchfilm.com/news/2011/11/screen-gems-takes-remake-rights-to-gareth-evans-the-raid.php |title=Screen Gems Takes Remake Rights to Gareth Evans' THE RAID |publisher=Twitchfilm.com |date=2011-11-09 |last=Brown |first=Todd |access-date=2012-03-26 |url-status=dead |archive-url=https://web.archive.org/web/20120326114927/http://twitchfilm.com/news/2011/11/screen-gems-takes-remake-rights-to-gareth-evans-the-raid.php |archive-date=26 March 2012}}&lt;/ref&gt; Screen Gems also wants the same choreographers from ''The Raid'' involved with the remake.&lt;ref name=&quot;collider&quot;/&gt; On 21 February 2014, the studio picked Patrick Hughes to direct the remake.&lt;ref&gt;{{cite web |title='Expendables 3' Director Boards 'The Raid' Remake |url=https://variety.com/2014/film/news/expendables-3-raid-screen-gems-1201102289/ |last=Kroll |first=Justin |work=Variety |date=2014-02-21}}&lt;/ref&gt; A day later, reports stated that both [[Chris Hemsworth|Chris]] and [[Liam Hemsworth]] were being eyed for roles by the studio.&lt;ref&gt;{{cite web |title=Hemsworth Brothers Sought For 'The Raid' Remake as 'The Expendables 3' Helmer Signs Up to Direct |url=http://blogs.indiewire.com/theplaylist/hemsworth-brothers-sought-for-the-raid-remake-as-the-expendables-3-helmer-signs-up-to-direct-20140220 |last= Jagernauth |first=Kevin |work=India Wire |date=2014-02-22}}&lt;/ref&gt; On 27 May 2014, ''Variety'' reported that the film's production was delayed until early 2015.&lt;ref&gt;{{cite web |url=https://variety.com/2014/film/news/screen-gems-delays-production-on-raid-remake-1201190302/ |title=Screen Gems Delays Production on 'Raid' Remake (EXCLUSIVE) |last=Kroll |first=Justin |work=Variety |date=27 May 2014 |access-date=27 May 2014}}&lt;/ref&gt; On 13 June 2014, [[Frank Grillo]] was the first to be announced to star in the remake, and is a fan of the original.&lt;ref&gt;{{cite web |url=http://wegotthiscovered.com/movies/frank-grillo-officially-cast-in-the-raid-redemption-remake/ |title=EXCLUSIVE: Frank Grillo Officially Cast in The Raid: Redemption Remake |last=Donato |first=Matt |work=We Got This Covered |date=12 June 2014 |access-date=12 June 2014}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://collider.com/captain-america-3-crossbones-frank-grillo/ |title=Frank Grillo Talks KINGDOM, His Involvement in THE RAID Remake, the Potential for More Rumlow / Crossbones in CAPTAIN AMERICA 3, and More |last=Nemiroff |first=Perri |work=Collider |date=13 June 2014 |access-date=13 June 2014}}&lt;/ref&gt; On 16 June 2014, Geek Tyrant revealed that the remake will be set in the near future.&lt;ref&gt;{{cite web |url=http://geektyrant.com/news/frank-grillo-cast-in-the-raid-remake |title=Frank Grillo Cast in The Raid remake |last=Paur |first=John |work=Geek Tyrant |date=16 June 2014 |access-date=16 June 2014}}&lt;/ref&gt; On 4 August 2014, ''TheWrap'' reported that [[Taylor Kitsch]] has been cast in the lead role, and also that XYZ Films is returning to produce the remake, which is expected to hew closely to the original film.&lt;ref&gt;{{cite web |url=http://www.thewrap.com/taylor-kitsch-offered-lead-in-the-raid-remake-exclusive/ |title=Taylor Kitsch Offered Lead in 'The Raid' Remake (Exclusive) |last=Sneider |first=Jeff |work=TheWrap |date=4 August 2014 |access-date=4 August 2014}}&lt;/ref&gt; On 22 August 2014, Hughes revealed that describes his version as being in the vein of ''[[Black Hawk Down (film)|Black Hawk Down]]'' and ''[[Zero Dark Thirty]]'' with 12 integral roles.&lt;ref&gt;{{cite web |title=Legendary Status Not Required: How Sylvester Stallone Plucked 'Expendables 3' Director Patrick Hughes Out of Obscurity |url=http://grantland.com/hollywood-prospectus/legendary-status-not-required-how-sylvester-stallone-plucked-expendables-3-director-patrick-hughes-out-of-obscurity/ |last=Patches |first=Matt |work=Grantland |date=2014-08-15}}&lt;/ref&gt; According to Tracking Board, on 23 October 2015, both Screen Gems and Kitsch had dropped out of the project and Hughes also dropped out as director.&lt;ref&gt;{{cite web|url=http://www.tracking-board.com/exclusive-patrick-hughes-exits-the-raid-remake-along-with-screen-gems/ |title=Patrick Hughes Exits 'The Raid' Remake, Along With Screen Gems (Exclusive) |publisher=Tracking-Board.com |date=23 October 2015}}&lt;/ref&gt; In February 2017, XYZ Films revealed that [[Joe Carnahan]] will produce and direct the remake with Evans as a producer.&lt;ref&gt;{{cite web |title=Joe Carnahan and Frank Grillo Are Tackling 'The Raid' Remake |url=http://grantland.com/hollywood-prospectus/legendary-status-not-required-how-sylvester-stallone-plucked-expendables-3-director-patrick-hughes-out-of-obscurity/ |last=Chitwood |first=Adam |work=Collider |date=15 February 2017}}&lt;/ref&gt; On 10 January 2022, Deadline reported that the remake will premiere on [[Netflix]] with [[Patrick Hughes (filmmaker)|Patrick Hughes]] directing alongside [[Michael Bay]] producing and Evans executive produced.&lt;ref&gt;{{cite web |title=Michael Bay, Gareth Evans And Patrick Hughes Team On A New Reimagining Of Evans' 'The Raid' For Netflix|url=https://deadline.com/2022/01/michael-bay-gareth-evans-patrick-hughes-evans-the-raid-netflix-1234907004/ |last=Kroll |first=Justin |work=Deadline |date=January 10, 2022}}&lt;/ref&gt;

2382 In March 2007 he promised his nine-year-old son he would make an improbable comeback after watching the film ''[[Rocky Balboa (film)|Rocky Balboa]]'', starring [[Sylvester Stallone]], which was all about Rocky being older but making a ring comeback. In 2007 he achieved a silver medal at the European Powerlifting championships,&lt;ref&gt;[http://www.bdfpa.co.uk/results/2007/2007%20Euro%20PL.pdf Europe] bdfpa.co.uk&lt;/ref&gt; and then bronze at the 2009 World Championships. In 2010 he finally achieved his dream of becoming World Powerlifting Champion at the age of 41. He again won the World Championships in 2011, 2013, 2015, 2019, and 2021. His daughter Hayley Rutherford and his son Ryan Rutherford have also won their respective divisions at the WDFPF World Championships.

2383 * actor: [[John Travolta]] - [[Sylvester Stallone]] - [[Mark Hamill]]

2384 * actor: [[Sylvester Stallone]] - [[Michael J. Fox]] - [[Götz George]]

2385 * actor: [[Tom Cruise]] - [[Sylvester Stallone]] - [[Ralph Macchio]]

2386 * actor: [[Tom Cruise]] - [[Sylvester Stallone]] - [[Eddie Murphy]]

2387 * actor: [[Patrick Swayze]] - [[Sylvester Stallone]] - [[Eddie Murphy]]

2388 The chain is also involved in the emission of reality TV to deal with fashion and art. Anne Slowey, editor of a famous American women's magazine offers ''Stylista'', the subject of the recruitment of an assistant. ''The Janice Dickinson Modeling Agency'' reveals the castings made by the former wife of Sylvester Stallone and boss of a modeling agency, [[Janice Dickinson]]. Going against the foot of a beauty contest which it takes the form, the show ''In search of inner beauty'' is in competition of candidates who unbeknownst to them, their human qualities.

2389 *'''Delete''' without prejudice per [[WP:NotJustYet]]. The growing body of this actor's work is [[WP:V|verifiable]] and appears to push slightly at the instruction at [[WP:ENT]]. Toward the nominator's wish to direct readers to places outside these pages, [[WP:WAX|what exists]] in [[IMDB|a non-RS database]] is irrelevant to Wikipedia as we are not them... however, such databases allow editors hints to encourage a more diligent search... through which &quot;hint&quot; I was able to easily confirm this actor's character of Franklin Winthrop as being significant to [[Oz (TV series)]] [http://goliath.ecnext.com/coms2/gi_0199-2611053/Inside-the-Emerald-City-if.html] We can judge an actor by the most notable projects with which they had a role, or consider the length and depth of their careers, and we do our readers a disservice by looking only at the least and then declaring all as insignificant. Heck, even the most notable actor may have had small roles. IE: [[Sylvester Stallone]] had a large number of insignificant or uncredited roles in the beginning of his career... such as Stud, Extra at Wedding, Subway Thug No.1, Discothèque Patron, Youth in Park, Young Man in Crowd, Mafioso, Man on Street, etc. We do not judge by the least. '''[[User:MichaelQSchmidt|&lt;span style=&quot;color:blue;&quot;&gt;Schmidt,&lt;/span&gt;]]''' ''[[User talk:MichaelQSchmidt|&lt;sup&gt;&lt;small&gt;MICHAEL Q.&lt;/small&gt;&lt;/sup&gt;]]'' 03:52, 9 October 2011 (UTC)

2390 * ''[[Rocky]]'', a 1976 film directed by [[John G. Avildsen]], written by and starring [[Sylvester Stallone]]

2391 * ''[[Rocky II]]'', a 1979 film written and directed by and starring [[Sylvester Stallone]]

2392 In addition to the seven films for which he was nominated for the [[Academy Award for Best Sound Mixing]], [[Meteor (film)|''Meteor,'']] [[Fame (1980 film)|''Fame'']], [[Pennies from Heaven (1981 film)|''Pennies from Heaven'']], ''[[WarGames]],'' ''[[2010: The Year We Make Contact]]'', ''[[RoboCop]]'' and [[Total Recall (1990 film)|''Total Recall'']], Kohut was re-recording mixer for award-winning feature films [[Rich and Famous (1981 film)|''Rich and Famous'']]&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0082992/filmmakers|title=Rich and Famous - Cast {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; directed by the legendary [[George Cukor]], [[The Prince of Tides|''Prince of Tides'']]&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0102713/?ref_=recent_view_1|title=The Prince of Tides - Production &amp; Contact Info {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; directed by [[Barbra Streisand]], ''[[Dead Poets Society]]''&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0097165/?ref_=nm_filmo_pastfilmvid_10|title=Dead Poets Society - Production &amp; Contact Info {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; directed by [[Peter Weir]], and ''[[Basic Instinct]]''&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0103772/?ref_=nm_filmo_pastfilmvid_1|title=Basic Instinct - Production &amp; Contact Info {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; directed by [[Paul Verhoeven]]. Among his other feature film credits are fan favorites ''[[Rocky IV]]''&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0089927/?ref_=instant_tt_1&amp;q=rocky%20iv|title=Rocky IV - Production &amp; Contact Info {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; directed by Sylvester Stallone, ''[[Farewell to the King]]''&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0097334/?ref_=instant_tt_1&amp;q=farewell%20to%20the%20king|title=Farewell to the King - Production &amp; Contact Info {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; directed by [[John Milius]], ''[[The Presidio (film)|The Presidio]]''&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0095897/?ref_=nm_filmo_pastfilmvid_14|title=The Presidio - Production &amp; Contact Info {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; directed by [[Peter Hyams]], and [[The War of the Roses (film)|''The War of the Roses'']]&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0098621/?ref_=instant_tt_1&amp;q=the%20war%20of%20the%20roses|title=The War of the Roses - Production &amp; Contact Info {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; and ''[[Throw Momma from the Train|Throw Mama from the Train]]'',&lt;ref&gt;{{Cite web|url=https://pro.imdb.com/title/tt0094142/?ref_=instant_tt_1&amp;q=throw%20mama%20f|title=Throw Momma from the Train - Production &amp; Contact Info {{!}} IMDbPro|website=pro.imdb.com|access-date=August 29, 2019}}&lt;/ref&gt; both directed by [[Danny DeVito]].

2393 * [[Sylvester Stallone]]

2394 * Sylvester Stallone

2395 '''''The Expendables 2''''' is a 2012 American [[action film]] directed by [[Simon West]], written by [[Richard Wenk]] and [[Sylvester Stallone]] and based on a story by Ken Kaufman, David Agosto and Wenk. [[Brian Tyler (composer)|Brian Tyler]] returned to score the film. It is the sequel to ''[[The Expendables (2010 film)|The Expendables]]'' (2010), and is the second installment in ''[[The Expendables (film series)|The Expendables]]'' film series. The film stars an [[ensemble cast]] of largely action film actors consisting of Sylvester Stallone, [[Jason Statham]], [[Jet Li]], [[Dolph Lundgren]], [[Chuck Norris]], [[Terry Crews]], [[Randy Couture]], [[Liam Hemsworth]], [[Jean-Claude Van Damme]], [[Bruce Willis]], and [[Arnold Schwarzenegger]]. In the film, [[List of The Expendables characters#Main|The Expendables]] undertakes a mission which evolves into a quest for revenge against rival mercenary Jean Vilain, who murders one of their own men and threatens the world with a deadly weapon.

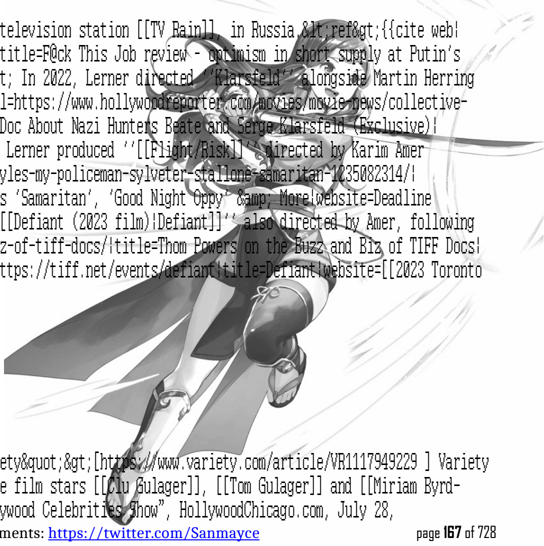
2396 [[File:The Expendables 2 Cast Roster.jpg|thumb|300px|Ensemble cast of action stars, including (left to right) Sylvester Stallone, Jason Statham, Jet Li, Dolph Lundgren, Chuck Norris, Jean-Claude Van Damme, Bruce Willis, Arnold Schwarzenegger and Terry Crews|alt=[[Photomontage]] of the film's stars]]

2397 * [[Sylvester Stallone]] as Barney Ross:

2398 On September 9, 2023, the film was made available in the Philippines for streaming on [[YouTube]] without charge by [[Viva Films]] to promote the upcoming release of ''[[Expend4bles]]'', slated for release in the country on September 20.&lt;ref&gt;{{cite web|author=[[Viva Films|VIVA Films]]|title=Expendables 2 Full Movie HD &amp;#124; Sylvester Stallone, Jason Statham, Jet Li, Dolph Lundgren, Chuck Norris|url=https://www.youtube.com/watch?v=3TtSUTHYYNE|website=[[YouTube]]|publisher=[[Google|Google LLC]]|access-date=September 11, 2023|date=September 9, 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite web|author=[[Viva Films|VIVA Films]]|title=Panibagong pelikula ba ang hanap niyo?|trans-title=Is it a new movie you're looking for?|url=https://www.facebook.com/VIVAFilms/posts/pfbid02joEczG4e1ivNPwTyTm4wYDdUqyLnseWFpoWKSdLtWWcj2h522yUM29Cig9uifgbQl|website=[[Facebook]]|publisher=[[Meta Platforms|Meta Platforms, Inc.]]|access-date=September 11, 2023|language=Filipino|date=September 9, 2023}}&lt;/ref&gt;

2399 &lt;ref name=&quot;CastStallone10&quot;&gt;{{cite web |url=http://www.blockbuster.co.uk/article/541984/sylvester-stallone-ignored-doctors-advice-to-do.htm |title=Sylvester Stallone ignored doctor's advice to do Expendables 2 stunts |publisher=[[Blockbuster LLC]] |date=August 15, 2012 |access-date=August 15, 2012

```
¦archive-url=https://web.archive.org/web/20130607040328/http://www.blockbuster.co.uk/article/541984/sylvester-stallone-ignored-doctors-advice-to-do.htm ¦archive-date=June 7, 2013 ¦url-status=dead }}&lt;/ref&gt;
2400 &lt;ref name=&quot;PG13&quot;&gt;{{cite news¦first=Pietro ¦last=Filliponi ¦url=http://www.dailyblam.com/news/2012/01/19/sylvester-stallone-confirms-the-expendables-2-will-be-rated-pg-13 ¦title=Sylvester Stallone Confirms The Expendables 2 Will be Rated PG-13 ¦publisher=The Daily Blam! ¦date=January 19, 2012 ¦access-date=January 19, 2012 ¦archive-url=https://archive.today/20130121085756/http://www.dailyblam.com/news/2012/01/19/sylvester-stallone-confirms-the-expendables-2-will-be-rated-pg-13 ¦archive-date=January 21, 2013 ¦url-status=live }}&lt;/ref&gt;
2401 &lt;ref name=&quot;VanDam1&quot;&gt;{{cite magazine¦last=Chris ¦first=Nashawaty ¦url=http://insidemovies.ew.com/2011/09/13/sylvester-stallone-expendables-2-exclusive/ ¦title=Sylvester Stallone talks about who’s on board for ‘Expendables 2’ - Exclusive ¦publisher=[[Time Inc.]] ¦magazine=[[Entertainment Weekly]] ¦date=September 13, 2011 ¦access-date=October 16, 2011 ¦archive-url=https://web.archive.org/web/20121008022958/http://insidemovies.ew.com/2011/09/13/sylvester-stallone-expendables-2-exclusive/ ¦archive-date=October 8, 2012 ¦url-status=live }}&lt;/ref&gt;
2402 &lt;ref name=&quot;Willis1&quot;&gt;{{cite web¦first=Jason ¦last=Barr ¦title=Sylvester Stallone Wants Bruce Willis to Play a &quot;Super Villain&quot; in The Expendables Sequel ¦website=[[Collider (website)¦Collider]] ¦url=https://www.collider.com/2010/08/29/the-expendables-sequel-sylvester-stallone-bruce-willis-super-villain/ ¦date=August 29, 2012 ¦access-date=November 16, 2012 ¦archive-url=https://web.archive.org/web/20120212053759/http://collider.com/the-expendables-sequel-sylvester-stallone-bruce-willis-super-villain/46297/ ¦archive-date=February 12, 2012 ¦url-status=dead }}&lt;/ref&gt;
2403 {{Sylvester Stallone}}
2404 [[Category:Films with screenplays by Sylvester Stallone]]
2405 ¦ [[Sylvester Stallone]]&lt;ref name=&quot;Stakar&quot; /&gt;
2406 &lt;ref name=&quot;Stakar&quot;&gt;{{Cite web ¦last=Peters ¦first=Megan ¦date=April 17, 2017 ¦title=Sylvester Stallone’s Mystery Role In Guardians Of The Galaxy Vol. 2 Revealed ¦url=https://comicbook.com/marvel/2017/04/17/guardians-of-the-galaxy-vol-2-sylvester-stallone-role/ ¦url-status=live ¦archive-url=https://web.archive.org/web/20170418013450/http://comicbook.com/marvel/2017/04/17/guardians-of-the-galaxy-vol-2-sylvester-stallone-role/ ¦archive-date=April 18, 2017 ¦website=[[ComicBook.com]]}}&lt;/ref&gt;
2407 &lt;ref name=&quot;OriginalGuardians&quot;&gt;{{Cite web ¦last=Daniell ¦first=Mark ¦date=April 25, 2017 ¦title=Sylvester Stallone’s ‘Guardians of the Galaxy Vol. 2’ character has a big future in the MCU ¦url=http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu ¦url-status=live ¦archive-url=https://web.archive.org/web/20170428051803/http://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu ¦archive-date=April 28, 2017 ¦access-date=April 28, 2017 ¦website=The Whig}}&lt;/ref&gt;
2408 In 2020, Knapp-Schwarzenegger (as a partner of GSB) merged with Stone, Genow, Smelkinson, &amp; Christopher, LLP, to form legal powerhouse GGSSC, LLP, which represents some of the top talent in Hollywood.&lt;ref&gt;{{Cite web ¦date=2021-08-30 ¦title=Jason Statham, Sylvester Stallone, 50 Cent, Megan Fox to Star in New ‘The Expendables’ Movie for Lionsgate (Exclusive) ¦url=https://www.hollywoodreporter.com/movies/movie-news/jason-statham-sylvester-stallone-50-cent-megan-fox-new-expendables-movie-cast-1235004992/ ¦access-date=2023-05-05 ¦website=The Hollywood Reporter}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦date=2022-08-30 ¦title=Nicolas Cage to Star in Comedy ‘Dream Scenario’ for A24 ¦url=https://www.thewrap.com/nicolas-cage-dream-scenario-comedy-a24-ari-aster/ ¦access-date=2023-05-05 ¦language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦last=Kroll ¦first=Justin ¦date=2022-08-03 ¦title=Conor McGregor To Make Acting Debut In Jake Gyllenhaal-Led ‘Road House’ For Prime Video ¦url=https://deadline.com/2022/08/conor-mcgregor-jake-gyllenhaal-road-house-1235083836/ ¦access-date=2023-05-05 ¦website=Deadline}}&lt;/ref&gt;
2409 After retiring from judo she lived in Paris and was once a bodyguard to [[Sylvester Stallone]] and [[Alain Delon]].&lt;ref name=JI/&gt;
2410 :::::Hmm - I spoke too soon - I just had to remove ''The Expendables 3'' (2015) from the [[Sylvester Stallone filmography]].  --[[User:Robsinden¦Rob Sinden]] ([[User talk:Robsinden¦talk]]) 15:20, 15 December 2011 (UTC)
2411 ¦ [[Saturday Night Live (season 23)¦23]] ¦¦ September 27, 1997 ¦¦ [[Sylvester Stallone]] ¦¦
2412 A satire of [[nightclub]]s and [[Clubbing (subculture)¦clubbing culture]]. It followed the exploits of brothers [[A Night at the Roxbury¦Doug and Steve Butabi]] (portrayed by [[Chris Kattan]] and [[Will Ferrell]] respectively), habitual clubbers dressed in [[rayon]] suits, as they attempt to pick up women on their outings. Their trademark was bobbing their heads in unison to the song &quot;[[What Is Love]]&quot; by [[Haddaway]], which always played throughout the duration of each sketch.&lt;ref name=Yang&gt;{{Cite web ¦last=Yang ¦first=Rachel ¦date=January 4, 2023 ¦title=The 12 best Will Ferrell sketches on ‘Saturday Night Live,’ from ‘More Cowbell’ to Harry Caray ¦url=https://ew.com/tv/2019/11/22/12-best-will-ferrell-sketches-saturday-night-live-more-cowbell-harry-caray/ ¦access-date=2024-05-19 ¦website=[[EW.com]] ¦language=en}}&lt;/ref&gt; The brothers were extremely unfortunate with the women at the clubs, often gyrating against them in an attempt to get them to dance, but always causing a negative reaction. They were frequently joined by a third person, often the [[List of Saturday Night Live hosts¦host]], who dressed and acted in a similar fashion (notably [[Tom Hanks]], [[Jim Carrey]], [[Sylvester Stallone]], [[Martin Short]], and [[Alec Baldwin]]); this character was usually credited as &quot;Barhop&quot;. Skits starring [[Pamela Anderson]] and [[Cameron Diaz]] were also featured. The sketch spawned the 1998 film ''[[A Night at the Roxbury]]''.&lt;ref name=&quot;Wild&quot; /&gt; Debuted March 23, 1996.
2413 ¦[[Sylvester Stallone]]
2414 ¦ [[Saturday Night Live (season 23)¦23]] ¦¦ September 27, 1997 ¦¦ [[Sylvester Stallone]] ¦¦
2415 ¦ ''[[The Expendables 2]]'' ¦¦ [[Lionsgate Films¦Lionsgate]] / [[Alchemy (company)¦Millennium Films]] / [[Nu Image]] ¦¦ [[Simon West]] (director), [[Sylvester Stallone]], [[Richard Wenk]] (screenplay); [[Sylvester Stallone]], [[Jason Statham]], [[Jet Li]], [[Dolph Lundgren]], [[Chuck Norris]], [[Jean-Claude Van Damme]], [[Bruce Willis]], [[Arnold Schwarzenegger]], [[Terry Crews]], [[Randy Couture]], [[Liam Hemsworth]], [[Scott Adkins]], [[Yu Nan]], [[Charisma Carpenter]], [[Amanda Ooms]], Nikolette Noel, Lyubomir Simeonov, ¦¦&lt;div style=&quot;text-align: center;&quot;&gt;&lt;ref&gt;{{cite web ¦url=https://comingsoon.net/films.php?id=75070 ¦title=The Expendables 2 ¦website=ComingSoon.net ¦access-date=2011-03-31}}&lt;/ref&gt;&lt;/div&gt;
2416 * [[Sylvester Stallone]] (2015)
2417 ** [[Sage Stallone]], actor, son of Sylvester Stallone (b. 1976)
2418 Also known for playing villainous characters, he appeared opposite [[Chuck Norris]] in ''[[The Cutter (film)¦The Cutter]]'' (2005), [[Jean-Claude Van Damme]] in ''[[Kill 'Em All (film)¦Kill 'Em All]]'' (2017), [[Sylvester Stallone]] in ''[[Escape Plan: The Extractors]]'' (2019), and [[Keanu Reeves]] in ''[[The Matrix Reloaded]]'' (2003), and ''[[John Wick (film)¦John Wick]]'' (2014). He also starred as Siro on the short-lived television series ''[[Mortal Kombat: Conquest]]'' (1998-99), based on the [[Mortal Kombat¦fighting video game of the same name]]. He has also worked as a stunt performer and fight choreographer on various productions, including ''[[Creed II]]'' (2018) and [[Nobody (2021 film)¦''Nobody'']] (2021).
2419 The premiere took place on July 1, 1991, at the [[Cineplex Odeon Corporation¦Cineplex Odeon]] in [[Century City]], Los Angeles.&lt;ref name=&quot;LATimesPremiere&quot; /&gt;&lt;ref name=&quot;GQPremiere&quot; /&gt; According to Fiedel, it was treated as a major event, unlike the premiere of ''The Terminator'', during which the audience was skeptical or laughed at the wrong times. Celebrities in attendance included [[Maria Shriver]], [[Nicolas Cage]], [[Sylvester Stallone]], [[Sharon Stone]], [[Michael Douglas]], and Furlong’s date [[Soleil Moon Frye]].&lt;ref name=&quot;TheRingerOral&quot;/&gt;&lt;ref name=&quot;GQPremiere&quot;/&gt;
2420 &lt;ref name=&quot;MensHealthLastAction&quot;&gt;{{Cite web ¦last=Ellis ¦first=Philip ¦date=June 9, 2019 ¦title=This ‘DeepFake’ Of Sylvester Stallone In Arnie’s Role As The Terminator Is Creepily Compelling ¦url=https://www.menshealth.com/entertainment/a27862500/sylvester-stallone-arnold-schwarzenegger-terminator/ ¦url-access=limited ¦url-status=live ¦archive-url=https://archive.today/20220106220817/https://www.menshealth.com/entertainment/a27862500/sylvester-stallone-arnold-schwarzenegger-terminator/ ¦archive-date=January 6, 2022 ¦access-date=January 6, 2022 ¦website=[[Men’s Health]]}}&lt;/ref&gt;
2421 * [[Arnold Schwarzenegger]] and [[Sylvester Stallone]] with Best Foreign Language Film
2422 * [[Sylvester Stallone]] (2015)
2423  ¦ ShortSummary= Follows [[Kenneth I. Starr¦Kenneth Starr]], who serves as an accountant to celebrities such as [[Sylvester Stallone]], [[Diane Sawyer]], and [[Wesley Snipes]]. Starr mismanages client’s money and steals millions of dollars, also marrying an [[exotic dancer]].&lt;ref&gt;{{cite web¦title=CNBC’S AMERICAN GREED SEASON 6 WILL PREMIERE ON WEDNESDAY, JANUARY 25TH ¦url=https://www.cnbc.com/id/45402280/CNBC_S_AMERICAN_GREED_SEASON_6_WILL_PREMIERE_ON_WEDNESDAY_JANUARY_25TH ¦publisher=CNBC.com ¦access-date=January 7, 2012 ¦format=Press Release ¦url-status=dead ¦archive-url=https://web.archive.org/web/20140811211632/http://www.cnbc.com/id/45402280/CNBC_S_AMERICAN_GREED_SEASON_6_WILL_PREMIERE_ON_WEDNESDAY_JANUARY_25TH ¦archive-date=August 11, 2014 }}&lt;/ref&gt;
2424 In 2021, Lerner produced ''F*ck This Job'' directed by Vera Krichevskaya, revolving around a woman deciding to open an independent television station [[TV Rain]], in Russia.&lt;ref&gt;{{cite web¦url=https://www.theguardian.com/film/2022/feb/22/fck-this-job-review-optimism-in-short-supply-at-putins-least-favourite-tv-channel¦title=F@ck This Job review - optimism in short supply at Putin’s least favourite TV channel¦website=The Guardian¦first=Phil¦last=Hoad¦date=February 22, 2022¦access-date=August 20, 2023}}&lt;/ref&gt; In 2022, Lerner directed ''Klarsfeld'' alongside Martin Herring revolving around [[Beate Klarsfeld]] and [[Serge Klarsfeld]], who confronted and hunted Nazi war criminals.&lt;ref&gt;{{cite web¦url=https://www.hollywoodreporter.com/movies/movie-news/collective-director-boards-nazi-hunter-documentary-klarsfeld-as-producer-4169101/¦title=‘Collective’ Director Alexander Nanau to Exec Produce Doc About Nazi Hunters Beate and Serge Klarsfeld (Exclusive)¦website=The Hollywood Reporter¦first=Alex¦last=Ritman¦date=April 20, 2021¦access-date=August 20, 2023}}&lt;/ref&gt; That same year, Lerner produced ''[[Flight/Risk]]'' directed by Karim Amer revolving around [[Boeing 737]] crashes.&lt;ref&gt;{{cite web¦url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/¦title=Prime Video Dates Fall Movie Slate: Harry Styles’ ‘My Policeman’, Lena Dunham’s ‘Catherine Called Birdy’, Sylvester Stallone’s ‘Samaritan’, ‘Good Night Oppy’ &amp; More¦website=Deadline Hollywood¦first=Anthony¦last=D’Alessandro¦date=August 1, 2022¦access-date=August 20, 2023}}&lt;/ref&gt; In 2023, Lerner produced ''[[Defiant (2023 film)¦Defiant]]'' also directed by Amer, following [[Disinformation in the Russian invasion of Ukraine]].&lt;ref&gt;{{cite web¦url=https://povmagazine.com/thom-powers-the-buzz-and-biz-of-tiff-docs/¦title=Thom Powers on the Buzz and Biz of TIFF Docs¦website=POV Magazine¦first=Pat¦last=Mullen¦date=August 9, 2023¦access-date=August 20, 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=https://tiff.net/events/defiant¦title=Defiant¦website=[[2023 Toronto International Film Festival]]¦access-date=August 20, 2023}}&lt;/ref&gt;
2425 ¦ [[Sylvester Stallone]], [[Chris Byrne (musician)¦Christopher Byrne]]
2426 ¦ [[Sylvester Stallone]], [[Chris Brown (American singer)¦Chris Brown]] &amp; [[Bow Wow (rapper)¦Bow Wow]]
2427 ¦ [[Sylvester Stallone]], [[Gabourey Sidibe]]
2428 ¦Guests=[[Sylvester Stallone]], [[Bill Bellamy]]
2429 ¦Guests=[[Sylvester Stallone]], [[Ben Foster (actor)¦Ben Foster]], [[Solomon Burke]]
2430 ¦Guests=[[Sylvester Stallone]], [[Carla Gugino]]
2431 '''''Vic''''' is a 2006 American [[short film]] [[drama]]. It is the directorial debut of [[Sage Stallone]].&lt;ref name=&quot;Variety&quot;&gt;[https://www.variety.com/article/VR1117949229 ] Variety “Four more for Boston fest: “Renaissance&quot;, “Vic,” others added to Sept. Event” by Jeff Sneider, August 29, 2006&lt;/ref&gt; The film stars [[Clu Gulager]], [[Tom Gulager]] and [[Miriam Byrd-Nethery]] with cameos by [[Carol Lynley]],&lt;ref name=&quot;Chicago&quot;&gt;“Interviews: George Kennedy, Carol Lynley at the Hollywood Celebrities Show”, HollywoodChicago.com, July 28,
```



2010&lt;/ref&gt; [[John LaZar]], and [[John Phillip Law]]. Stallone won the 2006 [[Boston Film Festival]] "Best New Filmmaker" award for the film.&lt;ref name=&quot;Variety&quot;/&gt;&lt;ref name=&quot;MovieFreak&quot;&gt;[http://www.moviefreak.com/features/bostonfilmfestival06.htm] Movie Freak "The 22nd Annual Boston Film Festival: Highlights from the 2006 Film Festival" by Gregory L. Amato&lt;/ref&gt; The film had its world premiere at the 2006 [[Palm Springs International Festival of Short Films]], where the cast and filmmakers were in attendance.&lt;ref name=&quot;PSFilmFest&quot;&gt;{{cite web¦url=http://www.psfilmfest.org/news/detail.aspx?NID%3D79%26year%3D2006 ¦title=2006 PALM SPRINGS INTERNATIONAL FESTIVAL OF SHORT FILMS &amp; SHORT FILM MARKET ANNOUNCES FESTIVAL WINNERS &amp;#124; News ¦accessdate=March 14, 2012 ¦url-status=dead ¦archiveurl=https://web.archive.org/web/20131021011320/http://www.psfilmfest.org/news/detail.aspx?NID=79&amp;year=2006 ¦archivedate=October 21, 2013 }} Palm Springs International Film Society web site "2006 PALM SPRINGS INTERNATIONAL FESTIVAL OF SHORT FILMS &amp; SHORT FILM MARKET ANNOUNCES FESTIVAL WINNERS", September 30, 2006&lt;/ref&gt;&lt;ref name=&quot;PalmSprings&quot;&gt;[https://www.imdb.com/title/tt0180283/releaseinfo] Palm Springs Film Festival&lt;/ref&gt;&lt;ref name=&quot;Squidwho&quot;&gt;"Sylvester Stallone Bio: Sylvester Stallone's Son Sage" Squidoo.com&lt;/ref&gt;

2432 * [[Sylvester Stallone]]

2433 ¦''''[[The Expendables (2010 film)¦The Expendables]]''''¦¦[[Sylvester Stallone]] ¦¦Sylvester Stallone, [[Jason Statham]], [[Jet Li]], [[Dolph Lundgren]], [[Mickey Rourke]]¦¦United States¦¦War adventure&lt;ref&gt;{{cite web¦title=The Expendables &gt; Overview¦work=[[AllMovie]]¦url=https://allmovie.com/work/the-expendables-474338¦author= Buchanan, Jason¦access-date=August 14, 2010}}&lt;/ref&gt;

2434 *[[Sylvester Stallone]] (1997)

2435 ¦Guests=[[Sylvester Stallone]], [[Loretta Lynn]]

2436 In March 2012, American actor [[Sylvester Stallone]] expressed interest in making an American adaptation of the film.&lt;ref&gt;{{cite news¦url=http://www.macleans.ca/general/on-feuding-with-arnold-schwarzenegger-and-whats-wrong-with-todays-action-heroes/¦title=In conversation: Sylvester Stallone¦last=Johnson¦first=Brian D.¦work=[[Maclean's]]¦date=2012-03-06¦accessdate=2015-03-19}}&lt;/ref&gt;

2437 From the film ''Rocky Balboa'' (2006), where Rocky (Sylvester Stallone) gives his only son (Milo Ventimiglia) some profound advice. [[User:Bulldog73¦Bulldog73]] [[User talk:Bulldog73¦&lt;sup&gt;talk&lt;/sup&gt;]] [[Special:Contributions/Bulldog73¦&lt;small&gt;da contribs&lt;/small&gt;]] [[Special:Random¦&lt;sub&gt;go rando&lt;/sub&gt;]] 04:57, 17 March 2012 (UTC)

2438 In 2008, Doucett testified along with Shandling, [[Sylvester Stallone]], and others against [[Anthony Pellicano]] for illegal wiretapping and other crimes.&lt;ref&gt;{{cite news¦url=https://www.variety.com/article/VR1117982684?refCatId=22 ¦ work=Variety ¦ first1=Marc ¦ last1=Graser ¦ title=Doucett testifies at Pellicano trial ¦ date= March 19, 2008}}&lt;/ref&gt;&lt;ref name=&quot;ca.findacase.com&quot;/&gt;&lt;ref&gt;{{cite news¦url=https://www.nytimes.com/2006/03/13/movies/a-studio-boss-and-a-private-eye-star-in-a-bitter-hollywood-tale.html¦title=A Studio Boss and a Private Eye Star in a Bitter Hollywood Tale¦first1=David M.¦last1=Halbfinger¦first2=Allison Hope¦last2=Weiner¦work=The New York Times ¦date=13 March 2006¦publisher=¦via=NYTimes.com}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=https://reporter.blogs.com/thresq/2009/04/pellicano-linda-doucett-lawsuit.html¦title=Linda Doucett sues Anthony Pellicano, Brad Grey, Bert Fields¦website=reporter.blogs.com}}&lt;/ref&gt;

2439 *The clue directed teams to travel to Hollywood Boulevard, where they had to locate four stars on the Hollywood Walk of Fame using a photograph of four peoples' heads edited onto Mount Rushmore: [[Sylvester Stallone]], [[Michael Jackson]], [[Steven Spielberg]], and [[Meryl Streep]]. Once they had located all four and taken a picture of each, they would give them to a woman dressed as [[Marilyn Monroe]] in exchange for their next clue.

2440 * [[Sylvester Stallone]]

2441 ''''Escape Plan'''' is a 2013 American [[prison film¦prison]] [[action thriller film]] starring [[Arnold Schwarzenegger]] and [[Sylvester Stallone]], and co-starring [[Jim Caviezel]], [[50 Cent]], [[Vinnie Jones]], [[Vincent D'Onofrio]] and [[Amy Ryan]].&lt;ref name=&quot;The Guardian&quot;&gt;{{cite news¦title=Schwarzenegger and Stallone join forces for The Tomb¦url=https://www.theguardian.com/film/2012/feb/09/arnold-schwarzenegger-sylvester-stallone-tomb ¦ location=London ¦ work=The Guardian¦first=Henry¦last=Barnes¦date=February 9, 2012}}&lt;/ref&gt; It was directed by Swedish filmmaker [[Mikael Håfström]], and written by Miles Chapman and [[Jason Keller (playwright)¦Jason Keller]] (under the [[anagram]] [[pen-name]] 'Arnell Jesko').&lt;ref&gt;{{cite web¦url=http://twitchfilm.com/2013/10/review-escape-plan-is-escapist-fun.html¦title=Review: Escape Plan Is Escapist Fun¦publisher=Twitch Film¦first=Jason¦last=Gorber¦date=October 18, 2013¦access-date=March 8, 2014¦archive-url=https://web.archive.org/web/20160304001146/http://twitchfilm.com/2013/10/review-escape-plan-is-escapist-fun.html¦archive-date=March 4, 2016¦url-status=dead}}&lt;/ref&gt; The first film to pair up Stallone and Schwarzenegger as co-leads,&lt;ref&gt;{{cite web¦url=http://cinemalowdown.com/2013/10/movie-review-escape-plan.html¦title=Movie Review: Escape Plan¦publisher=Cinema Lowdown¦first=Chaz¦last=Lipp¦date=October 18, 2013}}&lt;/ref&gt; it follows Stallone's character Ray Breslin, a lawyer turned prison security tester who is incarcerated in the world's most secret and secure prison, and recruits fellow inmate Emil Rottmayer, portrayed by Schwarzenegger, to stage a breakout. The film is the first installment of the [[Escape Plan (film series)¦''Escape Plan'' film series]].

2442 * [[Sylvester Stallone]] as Ray Breslin / Anthony Portos, a former prosecutor, businessman and skilled [[structural engineer]] who is best known for being the world's best escape artist.

2443 On April 9, 2013, it was officially announced that the film had been pushed back to a September 13, 2013 release and the film's title had been changed from ''The Tomb'' to ''Escape Plan''.&lt;ref&gt;{{cite web ¦url=https://www.comingsoon.net/news/movienews.php?id=102685 ¦title=The Tomb is Now The Escape Plan ¦date=April 10, 2013 ¦publisher=ComingSoon.net ¦access-date=April 11, 2013 ¦archive-date=July 23, 2014 ¦archive-url=https://web.archive.org/web/20140723152934/http://www.comingsoon.net/news/movienews.php?id=102685 ¦url-status=dead }}&lt;/ref&gt; On July 18, 2013, a fan screening was held at the Reading Cinemas Gaslamp 15 at [[San Diego Comic-Con]], which Stallone and Schwarzenegger attended.&lt;ref&gt;{{cite web¦url=https://www.hollywoodreporter.com/movies/movie-news/comic-con-sylvester-stallone-arnold-582260/¦title=Comic-Con: Sylvester Stallone, Arnold Schwarzenegger to Appear at 'Escape Plan' Screening¦first=Rebecca¦last=Ford¦work=[[The Hollywood Reporter]] ¦date=July 9, 2013¦access-date=December 21, 2021¦url-status=live¦archive-date=December 21, 2021¦archive-url=https://web.archive.org/web/20211221115631/https://www.hollywoodreporter.com/movies/movie-news/comic-con-sylvester-stallone-arnold-582260/}}&lt;/ref&gt; The film was theatrically released in the United States on October 18, 2013.

2444 ''Escape Plan'' was met with mixed reviews from critics. [[Rotten Tomatoes]] gives the film a rating of 50%, based on reviews from 107 critics, with an average score of 5.4/10. The site's critical consensus states: &quot;As much fun as it is to see Sylvester Stallone and Arnold Schwarzenegger team up onscreen, ''Escape Plan'' fails to offer much more than a pale imitation of 1980s popcorn thrills.&quot;&lt;ref&gt;{{Rottentomatoes¦escape_plan¦Escape Plan}}&lt;/ref&gt; [[Metacritic]] gives the film a score of 49 out of 100, based on 33 reviews, indicating &quot;mixed or average reviews&quot;.&lt;ref&gt;{{Metacritic film¦title=Escape Plan}}&lt;/ref&gt; Audiences surveyed by [[CinemaScore]] gave the film a grade &quot;B+&quot;.&lt;ref&gt;{{cite web ¦url= https://www.cinemascore.com/publicsearch/index/title/ ¦title= Cinemascore :: Movie Title Search ¦website= Cinemascore.com ¦access-date= July 19, 2019 ¦archive-url= https://web.archive.org/web/20181220122629/https://www.cinemascore.com/publicsearch/index/title/ ¦archive-date= December 20, 2018 ¦url-status= dead }}&lt;/ref&gt;

2445 In April 2017, a third film entered the early stages of development with Stallone again signed on to reprise his role as Ray Breslin.&lt;ref&gt;{{Cite web¦url=https://www.comingsoon.net/movies/news/837111-escape-plan-3-set-with-sylvester-stallone-to-return#/slide/1¦title = Escape Plan 3 Set with Sylvester Stallone to Return¦date = April 12, 2017}}&lt;/ref&gt; The film, ''[[Escape Plan: The Extractors]],''&lt;ref&gt;{{Cite web ¦date=2017-09-20 ¦url=https://deadline.com/2017/09/escape-plan-3-dave-bautista-max-zhang-sylvester-stallone-sequel-1202173372/ ¦title='Escape Plan 3': Dave Bautista, Max Zhang &amp; More Join Sylvester Stallone For Sequel – UPDATE ¦first=Amanda ¦last=N'Duka }}&lt;/ref&gt; was released direct-to-video on July 2, 2019.

2446 * [[Sylvester Stallone filmography]]

2447 * ''[[Lock Up (1989 film)¦Lock Up]]'', another Sylvester Stallone movie set in a prison

2448 # {{Icon¦C}} [[Sylvester Stallone]]

2449 *''[[First Blood]]'' (1982). Refusing to use a [[Stunt performer¦stunt double]] for certain scenes, [[Sylvester Stallone]] consequently suffered numerous serious injuries during filming. Instances include him having several ribs broken performing the stunt himself three times in which [[John Rambo]] escapes being pursued from Deputy Art Galt ([[Jack Starrett]]) by jumping off a cliff landing on tree branches. He suffered bruises to his back after filming 19 takes of the jail scene where Galt clubbed Rambo with a nightstick. He seriously injured his hand, almost losing his thumb after failing to realize that his hand was on the top of a gunfire squib that went off filming the scene where Rambo eludes the guards in the abandoned mine shaft. During the scene where Rambo is chased by the police on the stolen motorcycle, the stunt driver for [[Brian Dennehy]], [[Bennie Dobbins]], suffered a lumbar compression fracture performing an eventually deleted take of the stunt where the police car jumps the railroad. Dobbins would die from a heart attack in 1988 while filming ''Red Heat''.&lt;ref name=&quot;Cracked&quot;/&gt;&lt;ref&gt;{{cite web ¦url=https://www.film89.co.uk/fact-files-no-5-first-blood-1982/ ¦title=Fact File - No.5 - First Blood (1982) ¦first=Neil ¦last=Gaskin ¦website=Film '89 ¦date=16 December 2017 ¦access-date=22 August 2019 ¦archive-date=22 August 2019 ¦ archive-url=https://web.archive.org/web/20190822044643/https://www.film89.co.uk/fact-files-no-5-first-blood-1982/ ¦url-status=live }}&lt;/ref&gt;

2450 *''[[Rocky IV]]'' (1985). Demanding a sense of realism in the boxing match between [[Rocky Balboa]] and [[Ivan Drago]], Sylvester Stallone and [[Dolph Lundgren]] agreed to legitimately spar with each other. Stallone was airlifted from Canada to St. John's Hospital in [[Santa Monica, California]], and placed on intensive care for eight days after Lundgren delivered a hard punch to his chest, causing his heart to swell and his blood pressure to exceed 200.&lt;ref name=&quot;AICN&quot;&gt;{{cite web ¦ url=https://www.aintitcool.com/node/30932 ¦ title=Stallone Interview With Ain't It Cool News ¦ publisher=Ain't It Cool News ¦ access-date=28 June 2013 ¦ archive-date=30 September 2018 ¦ archive-url=https://web.archive.org/web/20180930092542/http://www.aintitcool.com/node/30932 ¦ url-status=live }}&lt;/ref&gt;&lt;ref name=&quot;BBC&quot;/&gt;

2451 * ''[[The Expendables (2010 film)¦The Expendables]]'' (2010). [[Sylvester Stallone]] injured his neck while filming a fight scene with [[Stone Cold Steve Austin¦Steve Austin]]. He required a metal plate inserted into his neck during surgery.&lt;ref name=&quot;BBC&quot;&gt;{{cite news ¦url=http://news.bbc.co.uk/2/hi/8443066.stm ¦title=Sylvester Stallone injures neck in fight scenes ¦ publisher=[[BBC]] ¦date=6 January 2010 ¦access-date=28 June 2013}}&lt;/ref&gt;&lt;ref name=&quot;News.com.au&quot;/&gt;

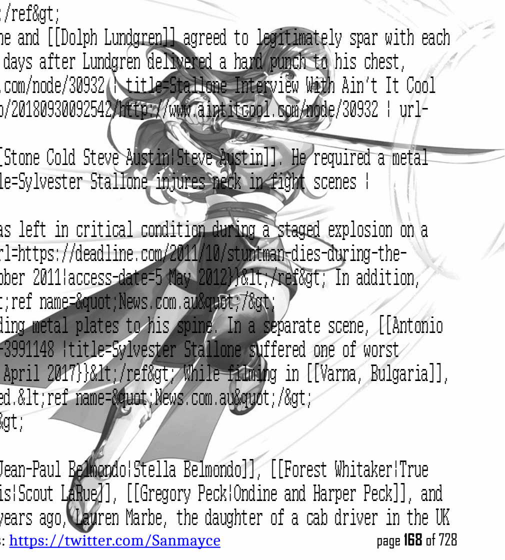
2452 * ''[[The Expendables 2]]'' (2012). While filming in [[Elin Pelin (town)¦Elin Pelin]], [[Bulgaria]], a stuntman was killed and another was left in critical condition during a staged explosion on a rubber boat. The surviving stuntman underwent a five-hour operation that reportedly left him in stable condition.&lt;ref&gt;{{cite web¦url=https://deadline.com/2011/10/stuntman-dies-during-the-filming-of-the-expendables-2/¦title=UPDATE: Stuntmen In 'Expendables 2' Fatal Accident Identified¦website=Deadline Hollywood¦date=31 October 2011¦access-date=5 May 2012}}&lt;/ref&gt; In addition, both Sylvester Stallone and [[Arnold Schwarzenegger]] required shoulder surgery after filming.&lt;ref name=&quot;ScreenSlam&quot;/&gt;&lt;ref name=&quot;News.com.au&quot;/&gt;

2453 *''[[The Expendables 3]]'' (2014). Sylvester Stallone suffered a serious back injury from a bad fall, requiring surgery that involved adding metal plates to his spine. In a separate scene, [[Antonio Banderas]] sustained a knee injury.&lt;ref&gt;{{cite news ¦url=https://www.mirror.co.uk/tv/tv-news/sylvester-stallone-suffered-one-worst-3991148 ¦title=Sylvester Stallone suffered one of worst injuries ever during filming of ''The Expendables 3'' ¦author=Greenwood, Carl ¦work=[[Daily Mirror]] ¦date=4 August 2014 ¦access-date=14 April 2017}}&lt;/ref&gt; While filming in [[Varna, Bulgaria]], a truck driven by [[Jason Statham]] lost its brakes and fell into the [[Black Sea]]. Statham was able to swim out of the accident unharmed.&lt;ref name=&quot;News.com.au&quot;/&gt;

2454 *[[Sylvester Stallone]]&lt;ref name=&quot;http://blogs.phillymag.com/the_philly_post/2012/09/28/celebrity-campaign-endorsements/&quot; /&gt;

2455 ¦quotee= Rocky(Slvester Stallone)

2456 Many others are daughters of actors and industry tycoons, such as [[Ava Philippe]], [[Jet Li¦Jane Li]], [[Ren Zhengfei¦Annabel Yao]], [[Jean-Paul Belmondo¦Stella Belmondo]], [[Forest Whitaker¦True Whitaker]], [[Warren Beatty¦Ella Beatty]], [[Mads Mikkelsen¦Viola Mikkelsen]], [[Sylvester Stallone¦Sophia-Rose Stallone]], [[Bruce Willis¦Scout LaRue]], [[Gregory Peck¦Ondine and Harper Peck]], and [[Sean Connery¦Natasha Connery]]. Others are famous in their own right, such as [[Ballet dancer¦ballerinas]] and young prodigies: a few years ago, Lauren Marbe, the daughter of a cab driver in the UK



whose IQ exceeded Einstein's, and in 2015, [[Olivia Hallisey]], an American high schooler who won the first prize of the [[Google Science Fair]] for inventing a test for the [[Ebola]] virus.&lt;ref&gt;{{cite web | url=https://madame.lefigaro.fr/celebrites/bal-des-debutantes-2018-qui-sont-les-debs-true-whitaker-gabrielle-de-pourtales-shloka-birla-2018-221118-151899 | title=Bal des débutantes : Qui sont les &quot;debs&quot; 2018 ? | date=23 November 2018 }}&lt;/ref&gt; In 2022 [[Eileen Gu]] attended le Bal&lt;ref&gt;{{Cite web |last=Noyen |first=Maria |title=I went to Le Bal des Débutantes, where kids of Hollywood legends, royals, and business moguls make their high-society debut. It was a night full of glamour I'll never forget. |url=https://www.insider.com/inside-le-bal-des-debutantes-attended-by-stars-and-royals-2022-12 |access-date=2023-03-17 |website=Insider |language=en-US}}&lt;/ref&gt; after winning two gold medals at the [[2022 Winter Olympics]].&lt;ref&gt;{{Cite web |date=2022-06-08 |title=Chinese Olympian Eileen Gu working for Salt Lake Games bid |url=https://apnews.com/article/eileen-gu-winter-olympics-sports-china-beijing-b1549a2c689f7086af97993dfcadadc3 |access-date=2023-03-17 |website=AP NEWS |language=en}}&lt;/ref&gt;

2457 * 2012: Sophia Rose Stallone in [[Elie Saab]] Haute Couture with [[Sylvester Stallone]]&lt;ref&gt;{{Cite web|title = Le Bal des Débutantes in Paris - Vogue.it|url = http://www.vogue.it/en/people-are-talking-about/parties-events/2012/11/le-bal-des-debutantes-in-paris|website = www.vogue.it|accessdate = 2015-10-15}}&lt;/ref&gt;

2458 The band signed to Epic Records in 1987, and recorded their first single - a [[Giorgio Moroder]] and [[Tom Whitlock]] track titled &quot;All I Need Is You&quot;. It was released as a promotional single, and featured in the film ''[[Over the Top (1987 film)|Over the Top]]'', starring [[Sylvester Stallone]].&lt;ref&gt;{{Cite web|url=http://www.imdb.com/title/tt0093692/soundtrack|title=Over the Top (1987) |website=IMDb.com}}&lt;/ref&gt; The song was included on the official soundtrack release, which was released during the same year.&lt;ref&gt;{{cite web|url=http://www.discogs.com/Various-Over-The-Top-Original-Motion-Picture-Soundtrack/release/544395 |title=Various - Over The Top (Original Motion Picture Soundtrack) (Vinyl, LP) at Discogs |website=Discogs.com |date= |accessdate=May 20, 2012}}&lt;/ref&gt;

2459 | [[Golden Raspberry Award for Worst Screen Combo|Worst Screen Couple]]{{efn|Shared with [[Brad Pitt]] and tied with [[Sylvester Stallone]] and [[Sharon Stone]] in ''[[The Specialist]]''}}

2460 :It seems he has recently died. Should his article be deleted and redirected to a section in his father's article? [[Sylvester Stallone]] I don't really care either way, but someone brought up his notabilty on the talk page as borderline by WP standards.--[[User:Canoe1967|Canoe1967]] ([[User talk:Canoe1967|talk]]) 14:36, 16 July 2012 (UTC)

2461 He made his directorial debut with ''[[Kambakkht Ishq]]'', a 2009 Indian Hindi-language romantic comedy film produced by [[Sajid Nadiadwala]]. The film was based on the 2002 Tamil film, ''[[Pammal K. Sambandam]]''. It featured [[Akshay Kumar]] and [[Kareena Kapoor]] in the lead roles, along with [[Aftab Shivdasani]] and [[Amrita Arora]] in supporting roles. Hollywood actors [[Sylvester Stallone]], [[Denise Richards]], [[Brandon Routh]], and [[Holly Valance]] appeared in cameos, playing themselves. The film was released on 3 July 2009.

2462 The gallery buys and sells mostly works by modern and contemporary artists. It also sells pieces at art fairs by artists such as [[Pablo Picasso]], [[Kurt Schwitters]], [[Fernand Léger]], [[Lyonel Feininger]], Robert and Sonia Delaunay, Stallone and Fernando Botero. The gallery works with the estates of [[Yves Klein]], [[Wifredo Lam]], [[Louise Nevelson]], and [[Robert Indiana]] among others. It also arranged for an exhibition of [[Sylvester Stallone]] at the Museum of Modern Art in Nice in 2015&lt;ref&gt;{{Cite news|url=https://news.artnet.com/art-world/celebrity-artist-sylvester-stallone-tames-crowd-glitzy-french-riviera-show-opening-299222|title=Celebrity Artist Sylvester Stallone's Show Opening - artnet News|date=2015-05-18|work=artnet News|access-date=2018-10-09|language=en-US}}&lt;/ref&gt; and hosts the artwork of fashion designer [[Karl Lagerfeld]] and the architect [[Richard Meier]].&lt;ref&gt;{{Cite news|url=https://www.artsy.net/article/editorial-the-mother-and-daughter-behind-galerie-gmurzynska|title=The Mother and Daughter Behind Galerie Gmurzynska|date=2013-11-14|work=Artsy|access-date=2018-10-09|language=en}}&lt;/ref&gt;

2463 Dizozza's family vacationed at [[Candlewood Lake]] in Connecticut. That led to some acting in children's commercials of DuRona studios and a brief appearance in a wedding scene as an extra in the [[Sylvester Stallone]] vehicle [[The Lords of Flatbush]].&lt;ref&gt;{{Cite web|title = The Lord's of Flatbush (1974)|url = https://www.imdb.com/title/tt0071772/|website = IMDb|access-date = 2015-12-02}}&lt;/ref&gt;

2464 | [[Sylvester Stallone]]

2465 | style=&quot;text-align: center;&quot; | &lt;ref&gt;{{Cite web |last=Cranswick |first=Amie |date=November 13, 2018 |title=Sylvester Stallone, Matthew Modine and Ryan Guzman star in new Backtrace trailer |url=https://www.flickeringmyth.com/2018/11/sylvester-stallone-matthew-modine-and-ryan-guzman-star-in-new-backtrace-trailer/ |access-date=May 16, 2021 |website=flickeringmyth.com}}&lt;/ref&gt;

2466 * [[Sylvester Stallone]]

2467 * '''[[Sylvester Stallone]]''' - ''[[Rocky]]'' (ex aequo)

2468 :I second Domins Vobisdu's recommendation above about watching dubbed movies and TV series, although in general I think you should steer clear of dubbed-to-German Hollywood movies - the problem being that dubbed TV shows and dubbed movies are handled completely differently by completely different people. Dubbed movies with big Hollywood stars are generally using a rather small cast of high-profile voice actors, most of whom regularly dub several actors (the German voice of William Shatner is the same as that of Sean Connery, Bruce Willis is the same voice actor as Sylvester Stallone etc) - meaning that most voices in German dubbed movies tend to sound the same, which I find seriously grating (this is probably not that much of a problem if you only watch a movie or two, but after a lifetime of watching dubbed movies on TV, I just can't stand the standard voice actors' voices any more). In TV shows, voice actors are generally selected by their ability to match the original actor's voice which leads to a much more varied and satisfying listening experience. In addition to Dominus Vobisdu's excellent Star Trek recommendation, you might also have a look at the German edition of the Simpsons - the voice actors are brilliant, and the translators generally did an impressive job of translating all the puns and little jokes. As far as German movies go, the best ones I've seen recently were the films based on the &quot;Brenner&quot; novels by [[Wolf Haas]] (''Komm süßer Tod'', ''Silentium'' and ''Der Knochenmann'') and of course ''Indien'', starring [[Josef Hader]] who also plays police inspector Brenner in the Wolf Haas movies. All of these are in rather heavy Austrian dialect, so they might be a bit difficult to understand if you're just beginnging to watch German movies. If you like absurdist, slightly pythonesque humor, try some of the [[Helge Schneider]] movies and the old eponymous TV series by [[Vicco von Bülow|Loriot]] - the latter especially is a great way to pick up on your language skills as part of the series' humor is that most everyone is talking in an incredibly well-pronounced, exact and extremely grammatically correct tone -- [[User:Ferkelparade|Ferkelparade]] [[User_talk:Ferkelparade|&amp;pi;]] 22:03, 23 July 2012 (UTC)

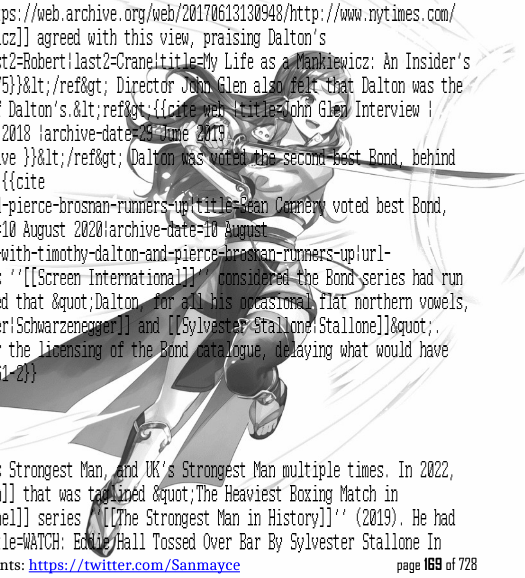
2469 According to Hindi dubbing director of [[Main Frame Software Communications]] [[Eliza Lewis|Ellie Lewis]], Samay Thakkar has been voicing various characters in many Hollywood films in [[Hindi language|Hindi]]. He is the Hindi dubbing voice of [[Christian Bale]] and [[Ben Affleck]]'s role as [[Batman|Bruce Wayne / Batman]] in the [[Christopher Nolan]]'s Batman reboot trilogy. He also voiced [[Arnold Schwarzenegger]] and [[Sylvester Stallone]]'s roles in all of his films as well.&lt;ref name=lm /&gt; He has also dubbed for [[Ajith Kumar]] and [[Vikram (actor)|Vikram]] in their [[Tamil language]] films and for [[Nagarjuna (actor)|Nagarjuna]] in his [[Telugu language]] films.{{citation needed|date=November 2019}}

2470 He played Nicky Goldberg in the independent film ''Songbird'' directed by Alex Boothby, Don Giovanni in ''Avenging Angelo'' ([[Sylvester Stallone]]/Warner Bros.), Young Anwyn Noble in ''[[Prince Charming (2001 film)|Prince Charming]]'' ([[Martin Short]]/Hallmark) and Bob Kachler in ''Rad'' ([[Lori Loughlin]]/Taliafilm).

2471 *[[Verne Lundquist]]&lt;ref&gt;{{YouTube|title=ABC Sports Soccer Bowl 80 Halftime with Pele and Sylvester Stallone Escape to Victory|id=7WF0NSvo510}}&lt;/ref&gt;

2472 * [[Sylvester Stallone]]

2473 [[Raymond Benson]] noted that Dalton &quot;purposely played Bond as a ruthless and serious man with very little of the wit displayed by Connery, Lazenby or Moore&quot;,{{sfn|Yeffeth|2006|p=10}} and considered him &quot;the most accurate and literal interpretation of the role&amp;nbsp;... ever seen on screen&quot;.{{sfn|Benson|1988|p=259}} His character also reflected a degree of moral ambiguity; in ''Licence to Kill'', for instance, he becomes a rogue agent,{{sfn|Spicer|2003|p=185}} while Dalton himself saw the character as a &quot;man, not a superhuman; a man who is beset with moral confusions and apathies and uncertainties, and who is often very frightened and nervous and tense&quot;.{{sfn|Benson|1988|p=259}} Smith and Lavington observed that during Dalton's portrayal in ''[[Licence to Kill]]'', Bond appeared &quot;self-absorbed&amp;nbsp;... reckless, brutal, prone to nervous laughter and&amp;nbsp;... probably insane, or at least seriously disturbed.{{sfn|Smith|Lavington|2002|p=225}} In the light of ''Licence to Kill'', one academic, Martin Willis, referred to Dalton's Bond as a &quot;muscular vigilante&quot;.{{sfn|Lindner|2009|p=169}} Steven Jay Rubin noted that Dalton's films had &quot;a hard-edged reality and some unflinching violent episodes that were better suited to Dalton's more realistic approach to the character&quot;.{{sfn|Rubin|2003|p=97}} Rubin considered Dalton's portrayal &quot;Fleming's Bond&amp;nbsp;... the suffering Bond&quot;.{{sfn|Rubin|2003|p=99}} In contrast to the previous incarnations of the character, Smith and Lavington identified Dalton's humour as &quot;brooding rather than flippant&quot;;{{sfn|Smith|Lavington|2002|p=213}} combined with his heavy smoking, they considered him &quot;an effective leading man&quot;.{{sfn|Smith|Lavington|2002|p=213}} Although Bond screenwriter [[Richard Maibaum]] called [[Sean Connery]] the best Bond, he considered Dalton the best actor of the four he worked with.&lt;ref&gt;{{cite web |first=Aljean |last=Harmetz |url=https://www.nytimes.com/1989/07/09/movies/film-creating-a-thriller-their-words-are-their-bond.html |title=Creating a Thriller, Their Words Are Their Bond |date=9 July 1989 |work=[[The New York Times]] |access-date=9 February 2017 |archive-date=13 June 2017 |archive-url=https://web.archive.org/web/20170613130948/http://www.nytimes.com/1989/07/09/movies/film-creating-a-thriller-their-words-are-their-bond.html |url-status=live }}&lt;/ref&gt; Screenwriter [[Tom Mankiewicz]] agreed with this view, praising Dalton's &quot;androgynous... and evil&quot; screen persona.&lt;ref&gt;{{cite book|first1=Tom|last1=Mankiewicz|author-link1=Tom Mankiewicz|first2=Robert|last2=Crane|title=My Life as a Mankiewicz: An Insider's Journey Through Hollywood|publisher=[[University Press of Kentucky]]|location=Lexington, Kentucky|date=2012|page=167|isbn=9780813140575}}&lt;/ref&gt; Director John Glen also felt that Dalton was the best actor who portrayed Bond, and that he was ahead of his time, noting the comparisons between Daniel Craig's incarnation to that of Dalton's.&lt;ref&gt;{{cite web |title=John Glen Interview | date=31 August 2017 |url=https://www.youtube.com/watch?v=fsIwjzHsESU |publisher=James Bond Radio Podcast #125 |access-date=4 December 2018 |archive-date=29 June 2019 |archive-url=https://web.archive.org/web/20190629043640/https://www.youtube.com/watch?v=fsIwjzHsESU&amp;gl=US&amp;hl=en |url-status=live }}&lt;/ref&gt; Dalton was voted the second-best Bond, behind Connery, in a publicly conducted poll by ''[[Radio Times]]'' in anticipation to the 25th Bond film, ''[[No Time to Die]]''.&lt;ref&gt;{{cite news|first=Catherine|last=Shoard|url=https://www.theguardian.com/film/2020/aug/10/sean-connery-voted-best-bond-with-timothy-dalton-and-pierce-brosnan-runners-up|title=Sean Connery voted best Bond, with Timothy Dalton and Pierce Brosnan runners up|newspaper=[[The Guardian]]|location=London, England|date=10 August 2020|access-date=10 August 2020|archive-date=10 August 2020|archive-url=https://web.archive.org/web/20200810181054/https://www.theguardian.com/film/2020/aug/10/sean-connery-voted-best-bond-with-timothy-dalton-and-pierce-brosnan-runners-up|url-status=live}}&lt;/ref&gt; Dalton's films did not perform as well at the box office as most of the previous films. Commentators such as ''[[Screen International]]'' considered the Bond series had run its course in the age of series such as [[Indiana Jones]] and ''Lethal Weapon''.{{sfn|Chapman|2009|p=211}} Edward P. Comentale observed that &quot;Dalton, for all his occasional flat northern vowels, was probably too much the stage actor to be convincing as an action hero in the age of [[Bruce Willis|Willis]], [[Arnold Schwarzenegger|Schwarzenegger]] and [[Sylvester Stallone|Stallone]]&quot;.{{sfn|Comentale|Watt|Willman|2005|p=139}} After just two films-''The Living Daylights'' and ''Licence to Kill''-litigation ensued over the licensing of the Bond catalogue, delaying what would have been Dalton's third film by several years. His six-year contract expired in 1993 and he left the series in 1994.{{sfn|Moore|2012|pp=161-2}}



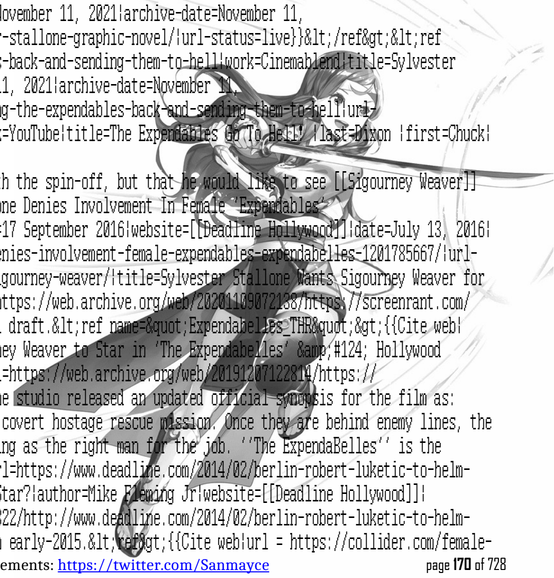
2474   <title>Category:Films with screenplays by Sylvester Stallone</title>

2475   <text bytes="334" sha1="i1pcl4qnozrm0c7mj2a5bq3iujxh5lg" xml:space="preserve">{{cat more|Sylvester Stallone}}

2476 [[Category:Works by Sylvester Stallone]]

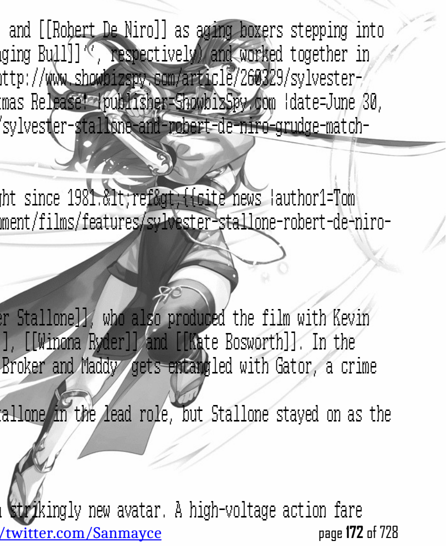
2477 Hall has won [[Strength athletics in the United Kingdom and Ireland|national competitions]] such as England's Strongest Man, Britain's Strongest Man, and UK's Strongest Man multiple times. In 2022, he was defeated by fellow World's Strongest Man winner [[Hafþór Júlíus Björnsson]] in a [[Eddie Hall vs. Hafþór Björnsson|boxing match]] that was taglined &quot;The Heaviest Boxing Match in History&quot;. He has presented his own television series called ''Eddie Eats America'' (2018) and was featured in the [[History Channel]] series ''[[The Strongest Man in History]]'' (2019). He had his first acting role as an extra in the action film ''[[Expend4bles]]'' (2023).&lt;ref&gt;{{cite news |last1=Patuto |first1=Greg |title=WATCH: Eddie Hall Tossed Over Bar By Sylvester Stallone In

```
'Expendables 4' Trailer |url=https://generationiron.com/expendables-4-trailer-eddie-hall-sylvester-stallone/ |work=Generation Iron Fitness &amp; Strength Sports Network |date=9 June 2023 |access-date=9 March 2024}}&lt;/ref&gt;
2478 [[File:Sylvester Stallone &amp; Bruce Willis (4840662146).jpg|thumb|upright|right|alt=Bruce Willis hugging fellow actor Sylvester Stallone|Willis hugging [[Sylvester Stallone]] in 2010]]
2479 | style=&quot;text-align:center;&quot; |&lt;ref&gt;{{cite web|url=https://www.independent.co.uk/arts-entertainment/films/news/greedy-and-lazy-all-too-expendable-bruce-willis-feels-wrath-sylvester-stallone-he-s-replaced-harrison-ford-expendables-3-8749641.html|title='Greedy and lazy'... all-too-Expendable Bruce Willis feels the wrath of Sylvester Stallone as he's replaced by Harrison Ford in Expendables 3|date=August 8, 2013|first=Adam|last=Sherwin|website=The Independent|archive-url=https://web.archive.org/web/20210610182501/https://www.independent.co.uk/arts-entertainment/films/news/greedy-and-lazy-all-too-expendable-bruce-willis-feels-wrath-sylvester-stallone-he-s-replaced-harrison-ford-expendables-3-8749641.html|archive-date=June 10, 2021|url-status=live|access-date=June 11, 2021}}&lt;/ref&gt;
2480 | style=&quot;text-align:center;&quot; |&lt;ref&gt;{{cite web|url=https://collider.com/expendables-2-movie-poster-bruce-willis/|title=12 Expendables 2 Posters Featuring Bruce Willis, Sylvester Stallone, Jason Statham, Chuck Norris, and More|date=April 26, 2012|first=Matt|last=Goldberg|website=Collider|archive-url=https://web.archive.org/web/20210610184012/https://collider.com/expendables-2-movie-poster-bruce-willis/|archive-date=June 10, 2021|url-status=live|access-date=June 11, 2021}}&lt;/ref&gt;
2481 |style=&quot;text-align:center;&quot;|&lt;ref&gt;{{cite news|url=https://www.hollywoodreporter.com/gallery/expendables-2-premiere-sylvester-stallone-arnold-schwarzenegger-363056/1-terry-crews|work=The Hollywood Reporter|accessdate=December 13, 2019|last=Godley|first=Chris|title='The Expendables 2' Premiere Arrivals: Sylvester Stallone, Arnold Schwarzenegger|date=August 16, 2012}}&lt;/ref&gt;
2482 |style=&quot;text-align:center;&quot;|&lt;ref&gt;{{cite news|url=https://www.hollywoodreporter.com/gallery/expendables-2-premiere-sylvester-stallone-arnold-schwarzenegger-363056/1-terry-crews|work=The Hollywood Reporter|accessdate=December 1, 2019|title='The Expendables 2' Premiere Arrivals: Sylvester Stallone, Arnold Schwarzenegger|last=Godley|first=Chris}}&lt;/ref&gt;
2483 Despite the contributions of former police officer Simon Michaël and [[Alphonse Boudard]], an acclaimed novelist known for both intimate and hardboiled crime stories, the writing process was not a smooth one. Deray opined that &quot;the story is simplistic and the dialogues are flat&quot;.&lt;ref name=&quot;mond-24-jul-20&quot;/&gt; Belmondo was also dissatisfied with the final script, which he felt was the primary reason for the film's failure, saying: &quot;There were writing issues that [it] was never able to leave behind.&quot; The original title was ''Cobra'' but, following the release of the eponymous [[Sylvester Stallone]] starrer, it was changed to ''The Loner'', whose similarity with ''The Outsider'' further highlighted its formulaic nature. Other considered titles were ''Règlements de compte'' ('Paybacks'), ''Superflic'' ('Supercop') and ''L'Ombre d'un flic'' ('Shadow of a Cop'), the latter being Deray and Belmondo's favorite.&lt;ref name=&quot;dur-bio&quot;/&gt;&lt;ref name=&quot;def-bel&quot;&gt;{{cite book |last1=Bourdon |first1=Laurent |date=November 15, 2017 |title=Définitivement Belmondo | chapter= |location=Paris |publisher=Larousse |pages= |isbn=9782035947765}}&lt;/ref&gt;
2484 * [[Sylvester Stallone]] (Krenn)
2485 [[File:Sylvester Stallone Cannes 2019.jpg|thumb|upright|Sylvester Stallone was the highest-paid actor in 1987, 1988, 1990, and 1995.]]
2486 | align=left | [[Sylvester Stallone]]
2487 | align=left | [[Sylvester Stallone]]
2488 | align=left | [[Sylvester Stallone]]
2489 | align=left | [[Sylvester Stallone]]
2490 | Best Supporting Actor || colspan=&quot;3&quot;| [[Sylvester Stallone]]&lt;br /&gt;''[[Creed (film)|Creed]]'' || [[Idris Elba]]&lt;br /&gt;''[[Beasts of No Nation (film)|Beasts of No Nation]]'' || colspan=&quot;2&quot;| [[Mark Rylance]]&lt;br /&gt;''[[Bridge of Spies (film)|Bridge of Spies]]''
2491 *''Stallone Sings'': Ralph sings in the voice of [[Sylvester Stallone]].
2492 When discussing current events with Smith, Garman will also perform impressions of famous actors such as [[Harrison Ford]], [[Al Pacino]], [[Charlton Heston]], [[Adam West]], [[Truman Capote]], [[Arnold Schwarzenegger]], [[Ed Wynn]], and [[Sylvester Stallone]]. Fans of the show have created an informal [[drinking game]] (mostly centered on Garman's phrase &quot;''How dare you, sir?''&quot; and Smith's overuse of the word &quot;''whatnot''&quot;), which the hosts have mentioned approvingly.
2493       <text bytes="40376" sha1="lp6gofbe3txngahoel33uw7r9ryemxn" xml:space="preserve">{{Short description|2010s action film series by Sylvester Stallone}}
2494 '''''The Expendables''''' is an American [[Ensemble cast|ensemble]] [[action thriller]] [[media franchise|franchise]] conceived by [[David Callaham]], spanning a film series, the first three ones cowritten by [[Sylvester Stallone]], and additional media. The films star an ensemble cast, notably Stallone and [[Jason Statham]], and are produced by [[Avi Lerner]] and Kevin King-Templeton. The film series, an acknowledgement of former blockbuster action films made in the 1980s and 1990s, also pays homage to action stars of former decades, and the more recent stars in action. The series consists of the films ''[[The Expendables (2010 film)|The Expendables]]'' (2010), ''[[The Expendables 2]]'' (2012), ''[[The Expendables 3]]'' (2014), ''[[Expend4bles]]'' (2023), and an ongoing comic book series, ''[[The Expendables Go to Hell]]'' (2021). Though criticism with regard to plot and dialogue between characters has been expressed, critics praised the use of comic relief in between action.
2495 | [[Sylvester Stallone]]
2496 | [[David Callaham]] &amp; Sylvester Stallone
2497 | [[Richard Wenk]] &amp; Sylvester Stallone
2498 | Katrin Benedikt, Sylvester Stallone &amp; Creighton Rothenberger
2499 | Sylvester Stallone
2500 In May 2012, it was announced that a television spin-off of the film series had entered development at Lionsgate.&lt;ref name=&quot;ExpendablesTV_Deadline&quot;&gt;{{Cite web|url = https://deadline.com/2012/05/lionsgate-considers-turning-the-expendables-into-a-tv-series-279898/|title = Lionsgate Considers Turning 'The Expendables' into a TV Series|date = May 31, 2012|access-date = August 13, 2019|archive-date = January 5, 2020|archive-url = https://web.archive.org/web/20200105201051/https://deadline.com/2012/05/lionsgate-considers-turning-the-expendables-into-a-tv-series-279898/|url-status = live}}&lt;/ref&gt; In March 2015, it was confirmed that the franchise would expand, with a television event series in active development. The project would be a joint-production between [[CBS Television Studios]], [[Lionsgate Television]], Shane Brennan Prods., and Rogue Marble Productions. [[Shane Brennan]] has signed on as [[Television program creator|creator]], [[Television producer|producer]] and [[showrunner]]; as well as serving as co-writer with [[Greg Coolidge]] and Kirk Ward. Stallone would serve as [[Television producer#Executive producer|executive producer]] alongside Avi Lerner, Kevin King, Grant Anderson, and Brennan. The plot of the series was announced to feature a new team of Expendables, composed of 'iconic television stars', who are on a mission to stop the dangerous activities of terrorist organizations.&lt;ref name=&quot;ExpendablesTV_Deadline2&quot;&gt;{{Cite web|url = https://deadline.com/2015/03/expendables-tv-series-fox-sylvester-stallone-1201386545/|title = 'The Expendables' Event Series in Works at Fox with Sylvester Stallone Producing|date = March 5, 2015|access-date = August 13, 2019|archive-date = February 24, 2020|archive-url = https://web.archive.org/web/20200224040731/https://deadline.com/2015/03/expendables-tv-series-fox-sylvester-stallone-1201386545/|url-status = live}}&lt;/ref&gt;
2501 | colspan=&quot;4&quot; | [[Sylvester Stallone]]
2502 * ''[[The Expendables Go to Hell]]'': Conceptualized during the writing of the [[The Expendables 2|second film]], the series was written by Chuck Dixon, from the original story co-written by [[Sylvester Stallone]] and Dixon. Based on a concept that Stallone had written for a movie he knew would never be produced, together with writing contributions from Richard C. Meyer, the series centers around the supernatural war that the titular mercenaries face after being killed and their adventures in [[hell]]. Together the team works to thwart [[Satan|the devil]] and his army of villainous minions (including [[Adolf Hitler]], [[Sadam Hussein]], [[Joseph Stalin]], and [[Osama bin Ladin]], among others). Funded through an online campaign, the series is published by Splatto Comics.&lt;ref name=&quot;Hell_BD&quot;&gt;{{cite web|url=https://bloody-disgusting.com/comics/3603123/sylvester-stallone-came-story-upcoming-graphic-novel-expendables-go-hell/|work=Bloody Disgusting|title=Sylvester Stallone Came Up With the Story for Upcoming Graphic Novel 'The Expendables Go to Hell'!|author=Squires, John|date=February 3, 2018|access-date=October 7, 2021|archive-date=October 8, 2021|archive-url=https://web.archive.org/web/20211008054727/https://bloody-disgusting.com/comics/3603123/sylvester-stallone-came-story-upcoming-graphic-novel-expendables-go-hell/|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Hell_DS&quot;&gt;{{cite web|url=https://www.digitalspy.com/movies/a30761388/expendables-cast-sylvester-stallone-graphic-novel/|work=Digital Spy|title=Sylvester Stallone teases &quot;next adventure&quot; of The Expendables|author=Seddon, Dan|date=April 20, 2020|access-date=November 11, 2021|archive-date=November 11, 2021|archive-url=https://web.archive.org/web/20211111154747/https://www.digitalspy.com/movies/a30761388/expendables-cast-sylvester-stallone-graphic-novel/|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Hell_CB&quot;&gt;{{cite web|url=https://www.cinemablend.com/news/2489624/sylvester-stallone-is-bringing-the-expendables-back-and-sending-them-to-hell|work=Cinemablend|title=Sylvester Stallone Is Bringing The Expendables Back And Sending Them To Hell|author=Evans, Nick|date=February 4, 2020|access-date=November 11, 2021|archive-date=November 11, 2021|archive-url=https://web.archive.org/web/20211111155145/https://www.cinemablend.com/news/2489624/sylvester-stallone-is-bringing-the-expendables-back-and-sending-them-to-hell|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Hell_CD&quot;&gt;{{cite AV media |url=https://www.youtube.com/watch?v=1AwV6XQW8VA|work=YouTube|title=The Expendables Go To Hell' |last=Dixon |first=Chuck|date=February 5, 2020|access-date=November 11, 2021}}&lt;/ref&gt;
2503 By February 2014, it was announced that [[Robert Luketic]] had been hired as director. Stallone stated that he is not involved with the spin-off, but that he would like to see [[Sigourney Weaver]] appear in the movie.&lt;ref name=&quot;Expendabelles_Deadline4&quot;&gt;{{cite news|last1=Fleming|first1=Mike Jr.|title=Sly Stallone Denies Involvement In Female 'Expendables' Spinoff|url=https://deadline.com/2016/07/sly-stallone-denies-involvement-female-expendables-expendabelles-1201785667/|access-date=17 September 2016|website=[[Deadline Hollywood]]|date=July 13, 2016|archive-date=November 22, 2019|archive-url=https://web.archive.org/web/20191122102009/https://deadline.com/2016/07/sly-stallone-denies-involvement-female-expendables-expendabelles-1201785667/|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Expendabelles_SR&quot;&gt;{{Cite web|url=https://screenrant.com/expendabelles-movie-sigourney-weaver/|title=Sylvester Stallone Wants Sigourney Weaver for 'ExpendaBelles' Movie|date=August 5, 2014|website=ScreenRant|access-date=March 8, 2020|archive-date=November 9, 2020|archive-url=https://web.archive.org/web/20201109072138/https://screenrant.com/expendabelles-movie-sigourney-weaver/|url-status=live}}&lt;/ref&gt; By August, Lerner stated that the script was nearing its final draft.&lt;ref name=&quot;Expendabelles_THR&quot;&gt;{{Cite web|url=https://www.hollywoodreporter.com/news/sylvester-stallone-wants-sigourney-weaver-723193|title=Sylvester Stallone Wants Sigourney Weaver to Star in 'The Expendabelles' &amp;#124; Hollywood Reporter|website=www.hollywoodreporter.com|date=August 4, 2014|access-date=March 8, 2020|archive-date=December 7, 2019|archive-url=https://web.archive.org/web/20191207122814/https://www.hollywoodreporter.com/news/sylvester-stallone-wants-sigourney-weaver-723193|url-status=live}}&lt;/ref&gt; Later that month, the studio released an updated official synopsis for the film as: &quot;''The Expendables'' universe has a new team. An elite group of highly trained female mercenaries are brought together for a covert hostage rescue mission. Once they are behind enemy lines, the women discover that they will also need to topple an evil dictator bent on world domination. It becomes clear there is no such thing as the right man for the job. ''The ExpendaBelles'' is the ultimate story of female empowerment and kick-ass teamwork.&quot;&lt;ref name=&quot;Expendabelles_Deadline5&quot;&gt;{{cite web|url=https://www.deadline.com/2014/02/berlin-robert-luketic-to-helm-millenniums-the-expendabelles-who-should-star/|title=Berlin: Robert Luketic To Helm Millennium's 'The ExpendaBelles'; Who Should Star?|author=Mike Fleming Jr|website=[[Deadline Hollywood]]|date=February 6, 2014|access-date=February 19, 2020|archive-date=July 7, 2014|archive-url=https://web.archive.org/web/20140707001322/http://www.deadline.com/2014/02/berlin-robert-luketic-to-helm-millenniums-the-expendabelles-who-should-star/|url-status=live}}&lt;/ref&gt; All studios involved had intended to begin production early-2015.&lt;ref&gt;{{Cite web|url = https://collider.com/female-
```



```
expendables-movie-update/|title = Producer Avi Lerner Provides Update on Female EXPENDABLES Movie, THE EXPENDABELLES; Plan to Begin Filming Early Next Year|website = [[Collider (website)|Collider]]|date = August 4, 2014|access-date = February 19, 2020|archive-date = January 9, 2020|archive-url = https://web.archive.org/web/20200109054056/https://collider.com/female-expendables-movie-update/|url-status = live}}&lt;/ref&gt; In November 2022, President of Millennium Films, Jeffrey Greenstein officially stated that the project was shelved. Stating that the project went through various iterations of having to &quot;explain&quot; why a team of only females was the center of the movie, the creatives have opted to instead integrate more female characters into the franchise.&lt;ref name=&quot;Future_THR&quot; /&gt;
2504 {{Sylvester Stallone}}
2505 * [[Sylvester Stallone]]&lt;ref name=&quot;FilmingBegin&quot; /&gt;
2506 | story          = Sylvester Stallone
2507 * [[Sylvester Stallone]]
2508 '''''The Expendables 3''''' is a 2014 American [[action film]] directed by [[Patrick Hughes (filmmaker)|Patrick Hughes]] and written by Creighton Rothenberger, Katrin Benedikt and [[Sylvester Stallone]]. It is the third installment in [[The Expendables (franchise)|''The Expendables'' franchise]] and the sequel to ''[[The Expendables (2010 film)|The Expendables]]'' (2010) and ''[[The Expendables 2]]'' (2012). The film features an [[ensemble cast]] of largely action film actors consisting of [[Jason Statham]], [[Sylvester Stallone]], [[Antonio Banderas]], [[Jet Li]], [[Wesley Snipes]], [[Dolph Lundgren]], [[Kelsey Grammer]], [[Randy Couture]], [[Terry Crews]], [[Kellan Lutz]], [[Ronda Rousey]], [[Glen Powell]], [[Victor Ortiz]], [[Mel Gibson]], [[Harrison Ford]], and [[Arnold Schwarzenegger]].
2509 * [[Sylvester Stallone]] as [[List of The Expendables characters#Barney Ross|Barney Ross]] &lt;br /&gt; The leader of the Expendables.&lt;ref&gt;{{cite web |url=http://www.lionsgatepublicity.com/epk/theexpendables2/docs/pro_notes.doc |publisher=[[Lions Gate Entertainment]] |work=Lions Gate Publicity |title=The Expendables 2 Production Notes |access-date=August 11, 2012 |format=DOC |archive-url=https://www.webcitation.org/6A8TbAlnf?url=http://www.lionsgatepublicity.com/epk/theexpendables2/docs/pro_notes.doc |archive-date=August 23, 2012 |url-status=dead |ref={{SfnRef|Production|2012}} |page=9}}&lt;/ref&gt;
2510 In March 2012, cast member [[Randy Couture]] said a third installment of ''[[The Expendables (film series)|The Expendables]]'' might begin production in late 2012 after the release of ''[[The Expendables 2]]''.&lt;ref name=&quot;Sequel1&quot; /&gt; In April 2012, [[Steven Seagal]] said he was offered a role in a third film.&lt;ref name=&quot;Sequel2&quot; /&gt; In August 2012, producer [[Avi Lerner]] confirmed that [[Nicolas Cage]] had been signed for the then-potential sequel. He also said that the producers intended to bring back the series' stars, attempting to have [[Mickey Rourke]] reprise his role, and had approached [[Clint Eastwood]], [[Harrison Ford]], and  [[Wesley Snipes]] to play new roles.&lt;ref name=&quot;Sequel3&quot; /&gt; [[Sylvester Stallone]] said &quot;We are thinking about different concepts-the third one is the hardest. The second is the natural progression. The third, that's when the air gets rare. We're thinking ambitiously about it, you now have to give audiences something they don't expect at all, Maybe, even going into a different genre&quot;.&lt;ref name=&quot;SequelAug15&quot; /&gt; Also in August, [[Chuck Norris]] said he would not return for a sequel.&lt;ref name=&quot;SequelNorris&quot; /&gt; On August 13, 2012, [[Jean-Claude Van Damme]] indicated that Stallone might include him in ''The Expendables 3'' as Claude Vilain, brother of Jean Vilain.&lt;ref name=&quot;SequelVanDamme&quot; /&gt; On October 31, 2012, it was confirmed that [[Nu Image]] and [[Millennium Films]] were in the process of pre-selling international distribution rights for ''The Expendables 3''.&lt;ref name=&quot;2012OctSequel&quot; /&gt; On December 19, 2012, it was reported that [[Jackie Chan]] had agreed to join the sequel on the condition that he would have more than a minor role.&lt;ref name=&quot;SequelChan&quot; /&gt; Chan later dropped out of the project due to scheduling conflicts.&lt;ref&gt;{{cite web|url=https://www.youtube.com/watch?v=0jewyFy3174|title=Expendables 3 - full press conference Mel Gibson Sylvester Stallone Jason Statham|last=moviemaniacsDE|date=August 13, 2014|via=YouTube|access-date=November 5, 2014|archive-date=April 25, 2017|archive-url=https://web.archive.org/web/20170425173626/https://www.youtube.com/watch?v=0jewyFy3174|url-status=live}}&lt;/ref&gt;
2511 On December 19, 2013, the first [[Trailer (promotion)|teaser trailer]] got released with the intent of showcasing the large cast of the movie.&lt;ref name=&quot;TeaserTrailer&quot; /&gt; At 2014 [[CinemaCon]], the official poster of the film was shown for the first time, although it only slightly differs from material shown before, featuring a white background instead of a black one.&lt;ref name=&quot;CinemaCon&quot; /&gt; The marketing scheme of putting a huge emphasis on the film's [[ensemble cast]] of famous action actors was continued in early April when 16 character posters of the film's cast were released and divided over four websites for maximum exposure.&lt;ref name=&quot;CharacterPosters&quot; /&gt; One day later, the first trailer with actual footage from the film premiered, again highlighting the film's huge cast, being referred to as a &quot;[[wikt:roll call|roll call]] trailer&quot; by [[Lionsgate]].&lt;ref name=&quot;Trailer1&quot; /&gt; At the 2014 Cannes Film Festival, [[Millennium Films]] hosted a special event promoting the film at the [[InterContinental Carlton Cannes Hotel|Carlton Hotel]] with [[Sylvester Stallone]], [[Jason Statham]], [[Arnold Schwarzenegger]], [[Mel Gibson]], [[Harrison Ford]], [[Wesley Snipes]], [[Antonio Banderas]], [[Dolph Lundgren]], [[Kelsey Grammer]], [[Randy Couture]], [[Kellan Lutz]], [[Victor Ortiz]], [[Glen Powell]] and director [[Patrick Hughes (director)|Patrick Hughes]] attending. The event lasted the entire day and featured the cast driving down the [[Croisette]] in tanks.&lt;ref name=&quot;Cannes&quot; /&gt;
2512 Unlike the first two films in the ''Expendables'' franchise, this film received a PG-13 rating instead of an R rating, which received backlash from the fans. At the [[2014 Cannes Film Festival]], [[Sylvester Stallone]] announced that he was aiming for a [[PG-13 (Motion Picture Association)|PG-13]] rating for the film. Stating that albeit it was close to being rated R, he wanted to reach a younger and broader audience with the new film.&lt;ref name=&quot;PG-13&quot; /&gt; However, when Stallone first submitted the film to the [[Motion Picture Association of America|MPAA]], they assigned it with the usual R rating and had to be trimmed down to be rated PG-13. On July 1, the MPAA granted ''The Expendables 3'' a PG-13 rating; the given description being &quot;violence including intense sustained gun battles and fight scenes and for language.&quot;
2513 Since its release, Stallone regrets the PG-13 rating, saying that it was &quot;a horrible miscalculation on everyone's part in trying to reach a wider audience but in doing such, diminish the violence that the audience expects. I'm quite certain it won't happen again.&quot;&lt;ref&gt;{{cite web |url=http://www.craveonline.com/film/interviews/792031-interview-sylvester-stallone-promises-r-rated-expendables-4 |title=Interview: Sylvester Stallone Promises R-Rated 'Expendables 4' |website=[[CraveOnline]] |date=November 23, 2014 |first=Fred |last=Topel |access-date=January 10, 2015 |archive-url=https://web.archive.org/web/20150219054148/http://www.craveonline.com/film/interviews/792031-interview-sylvester-stallone-promises-r-rated-expendables-4 |archive-date=February 19, 2015 |url-status=dead}}&lt;/ref&gt;
2514 One possible reason for this loss in ticket sales is because the film leaked three weeks ahead of its release; however, since most downloads were outside the United States, if every downloading American paid to see the movie, it would have made only $4 million more.&lt;ref&gt;{{cite news|url=https://www.nytimes.com/2014/08/18/movies/the-expendables-3-fumbles-its-mission.html|title='The Expendables 3' Fumbles Its Mission|first=Brooks|last=Barnes|newspaper=The New York Times|date=August 17, 2014|access-date=March 3, 2017|archive-date=December 6, 2017|archive-url=https://web.archive.org/web/20171206212051/https://www.nytimes.com/2014/08/18/movies/the-expendables-3-fumbles-its-mission.html|url-status=live}}&lt;/ref&gt; Another possible reason was the PG-13 cut which was meant to attract younger audiences. Only 34% of viewers were under 25 years old while the less violent cut may have reduced interest from older audiences.&lt;ref name=&quot;ForbesMendelson&quot; /&gt; [[Sylvester Stallone]] later admitted that toning the film down for a PG-13 rating was a mistake.&lt;ref&gt;{{cite web|url=https://www.cinemablend.com/new/Sylvester-Stallone-Finally-Admits-Expendables-Made-Huge-Mistake-68375.html|title=Sylvester Stallone Finally Admits The Expendables Made This Huge Mistake|date=November 24, 2014|access-date=April 17, 2020|archive-date=May 24, 2020|archive-url=https://web.archive.org/web/20200524095457/https://www.cinemablend.com/new/Sylvester-Stallone-Finally-Admits-Expendables-Made-Huge-Mistake-68375.html|url-status=live}}&lt;/ref&gt;
2515 * [[Sylvester Stallone filmography]]
2516 &lt;ref name=&quot;PG-13&quot;&gt;{{cite web |first=Kirsten |last=Acuna |title=Sylvester Stallone Explains Why 'The Expendables 3' Will Be Rated PG-13 Instead Of R |url=http://www.businessinsider.com/why-the-expendables-3-is-rated-pg-13-2014-5 |website=[[Business Insider]] |date=May 20, 2014 |access-date=May 21, 2014 |archive-date=February 13, 2015 |archive-url=https://web.archive.org/web/20150213160204/http://www.businessinsider.com/why-the-expendables-3-is-rated-pg-13-2014-5 |url-status=live }}&lt;/ref&gt;
2517 &lt;ref name=&quot;SequelMarch2013&quot;&gt;{{cite web |first=Dave |last=Trumbore |title=Sylvester Stallone Wants More Humor for Expendables 3, Hopes to Add Jackie Chan and Promises &quot;Young Blood&quot; |url=https://collider.com/sylvester-stallone-expendables-3/ |website=Collider |date=March 12, 2013 |access-date=March 13, 2013 |archive-url=https://web.archive.org/web/20130726152446/http://collider.com/sylvester-stallone-expendables-3 |archive-date=July 26, 2013 |url-status=live}}&lt;/ref&gt;
2518 {{Sylvester Stallone}}
2519 [[Category:Films with screenplays by Sylvester Stallone]]
2520 *... that the rights for '''''[[Rambo: The Video Game]]''''' were secured in part because of the scheduled release of the [[Sylvester Stallone]] action film ''[[The Expendables&amp;nbsp;2]]''?
2521 *[[Robert De Niro]] and [[Sylvester Stallone]] appear in the &quot;Three Wise Guys&quot; sketch.
2522 | caption = North American PC cover artwork featuring [[Sylvester Stallone]] as [[John Rambo]]
2523 Reef Entertainment acquired video game development and publishing rights in August 2011 to the [[Rambo (franchise)|''Rambo'' franchise]] from the rights holder StudioCanal.&lt;ref name=&quot;joystiq&quot;/&gt;&lt;ref name=&quot;mcv-2011&quot;&gt;{{cite web |last=Sacco |first=Dominic |date=August 4, 2011 |url=https://www.mcvuk.com/business/reef-reloads-rambo |title=Reef reloads Rambo |website=[[MCV (magazine)|MCV]] |publisher=[[NewBay Media]] |accessdate=April 3, 2018 |archive-date=August 2, 2023 |archive-url=https://web.archive.org/web/20230802101536/https://mcvuk.com/business-news/media-pr/reef-reloads-rambo/ |url-status=live }}&lt;/ref&gt; As of August 2011, the three ''Rambo'' series films had grossed over US$600 million.&lt;ref name=&quot;mcv-2011&quot;/&gt; The publisher said the purchase was fueled by the 2008 [[Rambo (2008 film)|''Rambo'']]'s success and by the then-scheduled release of the action film ''[[The Expendables 2]]'', starring [[Sylvester Stallone]].&lt;ref name=&quot;mcv-2011&quot;/&gt;&lt;ref&gt;{{cite web |last=Nicholson |first=Amy |date=August 15, 2012 |url=http://www.boxoffice.com/reviews/2012-08-the-expendables-2 |title=The Expendables 2 (Review) |website=[[BoxOffice (magazine)|BoxOffice]] |publisher=BoxOffice Media |archiveurl=https://web.archive.org/web/20120818002148/http://www.boxoffice.com/reviews/2012-08-the-expendables-2 |archivedate=August 18, 2012 |url-status=dead |accessdate=April 3, 2018}}&lt;/ref&gt; They stated that they planned to use the rights as a launching pad for the company.&lt;ref name=&quot;mcv-2011&quot;/&gt; In August 2011 the publisher said a ''Rambo: The Video Game'' would be released at retailers in 2012 and would be available on the [[PlayStation 3]], [[Xbox 360]], and [[Microsoft Windows]].&lt;ref name=&quot;mcv-2011&quot;/&gt; Jeffrey Matulef of ''[[Eurogamer]]'' said on October 5, 2012, that he did not believe the game would see release before the end of the year.&lt;ref&gt;{{cite web |last=Matulef |first=Jeffrey |date=October 5, 2012 |url=https://www.eurogamer.net/articles/2012-10-05-rambo-first-screenshot |title=New Rambo screenshot teases old locale |website=[[Eurogamer]] |publisher=Gamer Network |accessdate=December 21, 2012 |archive-date=February 13, 2022 |archive-url=https://web.archive.org/web/20220213140201/https://www.eurogamer.net/articles/2012-10-05-rambo-first-screenshot |url-status=live }}&lt;/ref&gt;
2524 Reef Entertainment's commercial director Craig Lewis describes the game as allowing players to &quot;get under the skin of Rambo and wield his iconic weapon-set in battle.&quot;&lt;ref&gt;{{cite web |last=Gallagher |first=Danny |date=July 6, 2012 |url=http://www.gametrailers.com/side-mission/20627/rambo-the-video-game-will-blow-stuff-up-at-gamescon |title=Rambo The Video Game Will Blow Stuff Up at Gamescom |website=[[GameTrailers]] |publisher=[[Viacom (2005-present)|Viacom]] |archiveurl=https://web.archive.org/web/20131010073205/http://www.gametrailers.com/side-mission/20627/rambo-the-video-game-will-blow-stuff-up-at-gamescon |archivedate=October 10, 2013 |url-status=dead |accessdate=April 3, 2018}}&lt;/ref&gt; The developer used the actors from the film series to provide voices in the game for both John Rambo, played by Sylvester Stallone, and Col. Trautman, played by [[Richard Crenna]].&lt;ref name=&quot;voice&quot;&gt;{{cite web |last=Fletcher |first=JC |date=March 1, 2013 |url=https://www.engadget.com/2013/03/01/rambo-the-video-game-has-stallone-and-crenna-voice-work-techni/ |title=Rambo: The Video Game has Stallone and Crenna voice work, technically |
```

```
website=Engadget (Joystiq) |publisher=Oath Inc. |accessdate=April 3, 2018 |archive-date=August 2, 2023 |archive-url=https://web.archive.org/web/20230802101531/https://www.engadget.com/2013-03-01-rambo-the-video-game-has-stallone-and-crenna-voice-work-techni.html |url-status=live }}&lt;/ref&gt; However, the developer did not actually hire the two actors to provide voice work (Crenna died in 2003) but instead contacted the owner of the film series, [[StudioCanal]], and acquired copies of the original voice tapes for the series to use in-game.&lt;ref name=&quot;voice&quot;/&gt;
2525 They produced also the ''Memoirs of Count Grammont'' (1793); ''The Economy of Human Life'' (1795) with plates by Gardiner from designs by Harding; [[Gottfried August Bürger]]'s ''[[Lenore (ballad)|Leonora]]'' (1796) translated by [[William Robert Spencer]]; and [[John Dryden]]'s ''Fables'' (1797), both illustrated with plates from drawings by [[Lady Diana Beauclerk]]. The first volume of their extensive series of historical portraits, ''The Biographical Mirrour'', with text by [[Francis Godolphin Waldron]], appeared in 1795. Silvester alone continued the ''Biographical Mirrour'', of which he issued the second volume in 1798; the third was ready for publication at the time of his death.
2526 *... that the rights for '''''[[Rambo: The Video Game]]''''' were secured in part because of the scheduled release of the [[Sylvester Stallone]] action film ''[[The Expendables&amp;nbsp;2]]''?
2527 :'''Easy verification:''' The hook can be verified from this line in the Development section: {{xt|The publisher said the purchase was fueled by the success of the 2008 Rambo film and by the scheduled release of the action film Expendables 2, starring Sylvester Stallone.}}
2528 :::*[[File:Symbol question.svg|16px]] Close paraphrasing issue has been successfully resolved, refs have been fixed, and release information has been clarified in a reasonable fashion, even if an actual release date does not seem to be currently available. The only thing I can see is missing is an inline source that carries the explicit information that ''The Expendables 2'' does indeed star Sylvester Stallone. The source given mentions the movie, but not specifically that it's a Stallone vehicle, just that it's an action movie. An extra citation tying Stallone to Expendables 2 should be easy enough to find, and in this case it is needed. Once you have that, this should be ready to go again. -[[User:BlueMoonset|BlueMoonset]] ([[User talk:BlueMoonset|talk]]) 05:18, 22 December 2012 (UTC)
2529 | ''[[Bullet to the Head]]'' || [[Warner Bros. Pictures]] / [[Dark Castle Entertainment]] / [[IM Global]] || [[Walter Hill]] (director); [[Alessandro Camon]] (screenplay); [[Sylvester Stallone]], [[Sung Kang]], [[Sarah Shahi]], [[Adewale Akinnuoye-Agbaje]], [[Christian Slater]], [[Jason Momoa]], [[Jon Seda]], [[Weronika Rosati]], [[Holt McCallany]], Dane Rhodes, Marcus Lyle Brown || style=&quot;text-align:center;&quot;| &lt;ref&gt;{{cite web |url=https://comingsoon.net/films.php?id=73900 |title=Bullet to the Head |website=ComingSoon.net |access-date=2012-04-07}}&lt;/ref&gt;
2530 | ''[[Escape Plan (film)|Escape Plan]]'' || [[Summit Entertainment]] / [[Emmett/Furla Oasis Films]] || [[Mikael Håfström]] (director); Miles Chapman, [[Jason Keller (playwright)|Jason Keller]] (screenplay); [[Sylvester Stallone]], [[Arnold Schwarzenegger]], [[Jim Caviezel]], [[50 Cent|Curtis &quot;50 Cent&quot; Jackson]], [[Vinnie Jones]], [[Vincent D'Onofrio]], [[Amy Ryan]], [[Sam Neill]], [[Faran Tahir]], [[Graham Beckel]], [[Matt Gerald]], [[Caitriona Balfe]] ||style=&quot;text-align:center;&quot;|  &lt;ref&gt;{{cite web |title=The Tomb |website=ComingSoon.net |url=https://comingsoon.net/films.php?id=43462 |access-date=2012-09-15}}&lt;/ref&gt;
2531 | ''[[Homefront (2013 film)|Homefront]]'' || [[Open Road Films]] / [[Alchemy (company)|Millennium Films]] / [[Nu Image]] || [[Gary Fleder]] (director); [[Sylvester Stallone]] (screenplay); [[Jason Statham]], [[James Franco]], [[Winona Ryder]], [[Kate Bosworth]], [[Frank Grillo]], [[Izabela Vidovic]], [[Chuck Zito]], [[Clancy Brown]], [[Rachelle Lefevre]], [[Omar Benson Miller]], [[Pruitt Taylor Vince]], [[Lance E. Nichols]], Marcus Hester, Austin Craig, Linds Edwards ||style=&quot;text-align:center;&quot;| &lt;ref&gt;{{cite web |title=Homefront |website=ComingSoon.net |url=https://comingsoon.net/films.php?id=93164 |access-date=2013-06-12}}&lt;/ref&gt;
2532 | ''[[Grudge Match]]'' || [[Warner Bros. Pictures]] || [[Peter Segal]] (director); [[Tim Kelleher (actor)|Tim Kelleher]], [[Rodney Rothman]] (screenplay); [[Sylvester Stallone]], [[Robert De Niro]], [[Alan Arkin]], [[Kevin Hart]], [[Kim Basinger]], [[Jon Bernthal]], [[LL Cool J]], [[Anthony Anderson]], [[Joey Diaz]], [[Barry Primus]], [[Don Lake]], [[Paul Ben-Victor]], [[Mykel Shannon Jenkins]], [[Greg Plitt]], [[Kate Reinders]], [[Griff Furst]], [[Frank Pesce]], [[Ireland Baldwin|Ireland Basinger Baldwin]], [[Jim Lampley]], [[Steve Levy]], [[John Buccigross]], [[Mike Goldberg]], [[Chael Sonnen]], [[Larry Merchant]], [[Roy Jones Jr.]], [[Michael Buffer]], [[Mike Tyson]], [[Evander Holyfield]], [[Manny Pacquiao]] ||style=&quot;text-align:center;&quot;| &lt;ref&gt;{{cite web |url=https://comingsoon.net/films.php?id=66310 |website=ComingSoon.net |title=Grudge Match |access-date=2013-01-22}}&lt;/ref&gt;
2533 |[[Sylvester Stallone]]
2534 In ''[[Cruel Intentions]]'', a dark-haired [[Sarah Michelle Gellar]] spreads the rumor that her rival has fake blonde hair.&lt;ref name=Page /&gt; In ''[[Rocky IV]]'', Sylvester Stallone battles a Russian blonde nemesis named Drago.&lt;ref name=Page /&gt; ''[[The Karate Kid]]'' features Johnny Lawrence; a wealthy, golden haired villain.&lt;ref name=Page /&gt; In ''[[Tangled]]'', the blond-haired Rapunzel takes revenge on her black-haired step mother.&lt;ref name=Page /&gt;
2535 ''Death Race 2000'' is a 1975 cult action film. It stars [[David Carradine]], [[Simone Griffeth]], [[Sylvester Stallone]], [[Louisa Moritz]], and [[Don Steele]], and was directed by [[Paul Bartel]]. In the near future, the ultimate sporting event is the Death Race. Contestants score points for running people down as they speed across the country. The sport has crazed fans who sacrifice themselves to the drivers. A covert group is trying to bring an end to the immoral Death Race and has infiltrated one of their followers into the race as a navigator of the top driver. In the end, the lives of the competitors, the President and the Death Race itself are in peril.
2536 | [[Sylvester Stallone]]
2537 This IP continues with vandalism. Edits have been reverted by bots and also manually. Today this IP inserted a fake notice about the death of Sylvester Stallone with went unnoticed for 1 hour. The IP has been blocked twice already, maybe it should be blocked longer or even forever. Thanks. -- [[User:80.139.244.182|80.139.244.182]] 16:55, 4 June 2006 (UTC)
2538 ** '''The Hammer''' (voiced by Eric Bauza impersonating [[Sylvester Stallone]]) is a mob enforcer who works for Don Vizioso. The Hammer has a mech suit that has a gripping claw, a large mallet, and rocket launchers which he wears when he goes into battle against any mutants. His soul was later drained by Kavaxas.
2539 In 2015, Coogler released his second film, ''[[Creed (film)|Creed]]'', a [[Spin-off (media)|spin-off]] of the [[Rocky (franchise)|''Rocky'' films]], which Coogler directed and co-wrote with [[Aaron Covington]]. The film starred [[Michael B. Jordan]] as [[Apollo Creed]]'s son [[Donnie Creed|Donnie]], who is trained and mentored by his father's old friend and former rival [[Rocky Balboa]], played by [[Sylvester Stallone]].&lt;ref name=&quot;Fleming&quot;&gt;{{cite web|last=Fleming|first=Mike|title='Fruitvale Station' Duo Ryan Coogler And Michael B. Jordan Team With Sly Stallone On MGM 'Rocky' Spinoff 'Creed'|url=https://www.deadline.com/2013/07/fruitvale-station-duo-ryan-coogler-and-michael-b-jordon-team-with-sly-stallone-on-mgm-rocky-spinoff-creed/|website=[[Deadline Hollywood]]|access-date=July 24, 2013|date=July 24, 2013}}&lt;/ref&gt;&lt;ref name=&quot;Rottenberg&quot;&gt;{{cite web |url=https://www.latimes.com/entertainment/movies/la-et-mn-rocky-creed-production-ryan-coogler-20151123-story.html|first=Josh |last=Rottenberg |title= Getting the 'Rocky' spinoff 'Creed' made was a real underdog story for director Ryan Coogler |work=Los Angeles Times|date=November 24, 2015|access-date=November 25, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://variety.com/2013/film/news/fruitvale-station-rocky-creed-mgm-1200567421/|title='Fruitvale Station' Team Eyeing 'Rocky' Spin-Off 'Creed' With MGM|last1=Kroll|first1=Justin|date=July 24, 2013|website=[[Variety (magazine)|Variety]]|access-date=June 30, 2015|quote=Sylvester Stallone is on board to reprise his role as Rocky Balboa, with Coogler penning the script along with Aaron Covington. Deadline Hollywood broke the news.}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|title = Michael B. Jordan Gives Millennials Their 'Rocky' With 'Creed'|url = https://www.nytimes.com/2015/11/01/movies/michael-b-jordan-gives-millennials-their-rocky-with-creed.html|newspaper =The New York Times|date =October 28, 2015|access-date =October 29, 2015|issn = 0362-4331|first = Cara|last = Buckley}}&lt;/ref&gt; It received critical acclaim from critics and audiences and grossed over $173 million worldwide. Among its accolades, Stallone won the National Board of Review Award for Best Supporting Actor, the [[Critics' Choice Movie Award for Best Supporting Actor]], and [[Golden Globe Award for Best Supporting Actor – Motion Picture]], and was nominated for the [[Academy Award for Best Supporting Actor]].&lt;ref name=&quot;Fleming&quot;/&gt;&lt;ref name=&quot;Rottenberg&quot;/&gt;
2540     <title>Category:Works by Sylvester Stallone</title>
2541       <comment>added [[Category:Sylvester Stallone]] using [[WP:HC|HotCat]]</comment>
2542       <text bytes="114" sha1="f8s334dvaw07naxqega413iv5bk6xmq" xml:space="preserve">{{Cat main|Sylvester Stallone}}
2543 [[Category:Sylvester Stallone]]</text>
2544 In March 2019, The Boies/Schiller Film Group filed suit on Tuesday, accusing Emmett/Furla Oasis Films of breaching a deal to produce  ''Escape Plan'' sequels starring [[Sylvester Stallone]]. According to the suit, Boies/Schiller put up a $6 million loan for the $65 million project. The arrangement also gave Boies/Schiller rights to finance and produce the sequels, ''[[Escape Plan 2: Hades]]'' and ''[[Escape Plan: The Extractors]]''. Boies/Schiller alleges that Emmett/Furla Oasis Films breached that agreement, and did not allow Boies/Schiller to put up financing for the sequels. The suit claims Emmett/Furla also failed to pay an executive producing fee on each of the two movies.&lt;ref&gt;{{cite news|url=https://variety.com/2019/film/news/boies-schiller-escape-plan-lawsuit-emmett-furla-1203178856/|title=Boies/Schiller Film Group Sues Over 'Escape Plan' Sequels |last=Maddaus |first=Gene |date=2019-04-02|work=Variety|language=en |df=mdy-all|access-date=2019-07-20}}&lt;/ref&gt;
2545 * [[Sylvester Stallone]]&lt;!--per poster block, De Niro is billed first--&gt;
2546 '''''Grudge Match''''' is a 2013 American [[sports comedy|sports comedy film]] directed by [[Peter Segal]]. The film stars [[Sylvester Stallone]] and [[Robert De Niro]] as aging boxers stepping into the ring for one last bout. Stallone and De Niro have both previously been in successful boxing films (''[[Rocky (franchise)|Rocky]]'' and ''[[Raging Bull]]'', respectively) and worked together in ''[[Cop Land]]''. It was previously scheduled for release on January 10, 2014, but was moved up to December 25, 2013.&lt;ref&gt;{{cite web |url=http://www.showbizspy.com/article/260329/sylvester-stallone-and-robert-de-niro-grudge-match-set-for-christmas-release.html |title=Sylvester Stallone and Robert De Niro: Grudge Match Set For Christmas Release! |publisher=ShowbizSpy.com |date=June 30, 2013 |access-date=July 8, 2013 |url-status=dead |archive-url=https://web.archive.org/web/20130709231701/http://www.showbizspy.com/article/260329/sylvester-stallone-and-robert-de-niro-grudge-match-set-for-christmas-release.html |archive-date=July 9, 2013 |df=mdy-all }}&lt;/ref&gt;
2547 * [[Sylvester Stallone]] as Henry &quot;Razor&quot; Sharp
2548 In preparation for the film, both actors trained with boxing trainer Bob Sale. During training, Stallone went down to 168 pounds, his lowest weight since 1981.&lt;ref&gt;{{cite news |author1=Tom Teodorczuk |title=Sylvester Stallone: Robert De Niro persuaded me to get back in the ring again |url=https://www.independent.co.uk/arts-entertainment/films/features/sylvester-stallone-robert-de-niro-persuaded-me-to-get-back-in-the-ring-again-9066643.html |access-date=26 March 2023 |work=[[The Independent]] |date=17 January 2014}}&lt;/ref&gt;
2549 | screenplay     = [[Sylvester Stallone]]
2550 * Sylvester Stallone
2551 '''''Homefront''''' is a 2013 American [[action film#Action-thriller|action thriller film]] directed by [[Gary Fleder]] and written by [[Sylvester Stallone]], who also produced the film with Kevin King Templeton and John Thompson. The film, which is based on the novel of the same name by Chuck Logan, stars [[Jason Statham]], [[James Franco]], [[Winona Ryder]] and [[Kate Bosworth]]. In the film, Phil Broker, a retired [[Drug Enforcement Administration|DEA]] agent, leads a peaceful life with his daughter Maddy, but trouble ensues as Broker and Maddy  gets entangled with Gator, a crime boss, due to a [[school bullying]] incident.
2552 [[Sylvester Stallone]] wrote the script, which is based on the novel of the same name by Chuck Logan. The film was originally intended to have Stallone in the lead role, but Stallone stayed on as the producer and passed on the lead role to [[Jason Statham]].&lt;ref name=&quot;variety-foundas&quot; /&gt;
2553 {{Sylvester Stallone}}
2554 [[Category:Films with screenplays by Sylvester Stallone]]
2555 [[Taran Adarsh]] of ''[[Bollywood Hungama]]'' gave 3.5/5 stars and wrote &quot;''Commando: A One Man Army'' is vintage good versus evil saga in a strikingly new avatar. A high-voltage action fare
```



that's racy, pulsating and packed with some adrenaline-pumping stunts.&quot;&lt;ref&gt;{{cite web|url=https://www.bollywoodhungama.com/movie/commando-a-one-man-army/critic-review/commando-a-one-man-army-movie-review/|title=Commando - A One Man Army Movie Review|website=[[Bollywood Hungama]]|date=12 April 2013 }}&lt;/ref&gt; Meena Iyer of ''[[The Times of India]]'' gave 3.5/5 stars and wrote &quot;If you are an action junkie and if you have peaked on [[Sylvester Stallone]]'s [[Rambo (franchise)|''Rambo''-series]], then our desi Commando, is what you will term paisa-vasool entertainment.&quot;&lt;ref&gt;{{cite news|url=https://timesofindia.indiatimes.com/entertainment/hindi/movie-reviews/commando-a-one-man-army/movie-review/19474087.cms|title=Commando - A One Man Army Movie Review|website=[[The Times of India]]}}&lt;/ref&gt;

2556 The movie generated some controversy when [[:pt:Carlos Alberto Prates Correia|Carlos Alberto Prates Correa]] called the movie &quot;fascist&quot; and &quot;right-wing&quot;. The distributor Manchete Video launched the VHS of the movie at the same time of the premiere. Brazilian [[Motion Picture Association|MPA]]'s Harry Stone compared the movie with [[Sylvester Stalone|Stalone]]'s [[Cobra (1986 movie)|Cobra]] in an attempt to criticize its violent content. There is an urban legend that dealers in Rio de Janeiro offered drugs in a package called &quot;capa preta&quot;, in homage to the film.&lt;ref&gt;http://estranhoencontro.blogspot.com/2015/03/o-homem-da-capa-preta.html {{Bare URL inline|date=August 2024}}&lt;/ref&gt;

2557 The novel's film rights were purchased in the 1980s by [[Sylvester Stallone]] to develop as a vehicle for himself but no film eventuated.&lt;ref name=Smith/&gt;&lt;ref&gt;{{cite news |last= Pond | first=Steve |title= Puttnam's Complaint |date= 13 November 1986 |work= The Washington Post |page= B9}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |title= Lakeland Ledger - Google News Archive Search|url= https://news.google.com/newspapers?nid=1346&amp;dat=19861124&amp;id=sWRNAAAAIBAJ&amp;sjid=lPsDAAAAIBAJ&amp;pg=6842,5083921&amp;hl=en }}&lt;/ref&gt;&lt;ref&gt;{{cite magazine|magazine=Filmink| first=Stephen|last=Vagg|url=https://www.filmink.com.au/the-cinema-of-wilbur-smith/|title=The Cinema of Wilbur Smith|date=January 27, 2022}}&lt;/ref&gt;

2558 * [[Sylvester Stallone]]

2559 '''''Reach Me''''' is a 2014 American [[drama film]] directed and written by [[John Herzfeld]].&lt;ref&gt;{{cite web|url=https://www.allmovie.com/movie/reach-me-v574900|publisher=[[AllMovie]]| title=Reach me|accessdate=February 3, 2015|last=Cooper|first=Tracie}}&lt;/ref&gt; The film stars [[Sylvester Stallone]], [[Kyra Sedgwick]], [[Terry Crews]], [[Thomas Jane]], [[Kevin Connolly (actor)| Kevin Connolly]], [[Lauren Cohan]], [[Kelsey Grammer]], and [[Tom Berenger]].&lt;ref&gt;{{cite web|url=http://www.digitalspy.co.uk/movies/news/a438116/sylvester-stallone-joins-indie-drama-reach-me.html|title=Sylvester Stallone joins indie drama 'Reach Me'|website=[[Digital Spy]]|date=November 14, 2012|accessdate=March 25, 2013}}&lt;/ref&gt;

2560 The film was produced by Rebekah Chaney, [[Cassian Elwes]], Buddy Patrick, John Herzfeld.&lt;ref&gt;{{cite magazine | last=Stedman | first=Alex | date=July 6, 2014 | title=Watch: Sylvester Stallone, Nelly, Kyra Sedgwick in First 'Reach Me' Trailer | url=https://variety.com/2014/film/news/watch-sylvester-stallone-nelly-kyra-sedgwick-in-first-reach-me-trailer-1201258429/ | magazine=[[Variety (magazine)|Variety]] | access-date=June 30, 2016}}&lt;/ref&gt;

2561 * [[Sylvester Stallone]] as Gerald

2562 During principal photography in 2013, funding for the film dried up when one of the investors, [[Norman Zada]], backed out and sued for return of {{US$|1000000}}.&lt;ref&gt;{{cite court | litigants=Norman Zada v. Rebecca Chaney et al. | pinpoint=Case No. BC508099 | court=[[Los Angeles Superior Court]] | url=https://unicourt.com/case/ca-la2-norman-zada-vs-rebecca-chaney-et-al-296592 | date=May 7, 2013 }}&lt;/ref&gt;&lt;ref&gt;{{cite web | date=May 8, 2013 | title=Movie Money | work=[[Courthouse News Service]] | url=https://www.courthousenews.com/movie-money-15/ | access-date=December 29, 2017}}&lt;/ref&gt; Herzfeld, Stallone and producers Rebekah Chaney and Cassian Elwes started a [[Kickstarter]] campaign to raise their goal of {{US$|250000}} by September 19.&lt;ref&gt;{{cite web| url=https://www.theverge.com/2013/8/21/4646040/sylvester-stallone-and-director-john-herzfeld-turn-to-kickstarter-reach-me| title=Sylvester Stallone and director John Herzfeld turn to Kickstarter to finish 'Reach Me'| author=Bryan Bishop |work=[[The Verge]] |date=August 21, 2013| accessdate=October 18, 2013}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://www.kickstarter.com/projects/reachme/reach-me-the-movie-kickstarter| title=REACH ME THE MOVIE@KICKSTARTER| author=John Herzfeld| publisher=[[Kickstarter]] |date=August 20, 2013 | accessdate=August 24, 2013}}&lt;/ref&gt; Despite reaching the {{US$|250000}} goal on Kickstarter, the production team decided to withdraw its Kickstarter campaign and start again with the competing crowdfunding platform [[Indiegogo]], citing its broader and more flexible capabilities.&lt;ref&gt;{{cite web| url=https://www.thewrap.com/sylvester-stallone-movie-reach-me-moves-from-kickstarter-to-indiegogo-exclusive/ |title=Sylvester Stallone Movie 'Reach Me' Boots Kickstarter for Indiegogo (Exclusive)| author=Jeff Sneider |publisher=The Wrap |date=September 18, 2013 |accessdate=October 18, 2013}}&lt;/ref&gt; The Indiegogo campaign set a goal for {{US$|50000}} starting on September 17 and ended on September 22 with a total of {{US$|178640}}.&lt;ref&gt;{{cite web| url=http://www.indiegogo.com/projects/reach-me-the-movie-at-indiegogo |title=REACH ME THE MOVIE AT INDIEGOGO |author=John Herzfeld |publisher=[[Indiegogo]] |date=September 17, 2013 | accessdate=October 18, 2013}}&lt;/ref&gt; Total production costs for the film are estimated at {{US$|5000000}}.&lt;ref name=wsj&gt;{{cite news | last=Steinberg | first=Don | date=November 21, 2014 | title=How Sylvester Stallone Helped Indie Film 'Reach Me' Get Made | url=https://www.wsj.com/articles/how-sylvester-stallone-helped-indie-film-reach-me-get-made-1416595119 | newspaper=[[The Wall Street Journal]] | access-date=June 29, 2016}}{{subscription required}}&lt;/ref&gt;

2563 The first official trailer for John Herzfeld's ''Reach Me'' was released on July 7, 2014,&lt;ref&gt;{{cite news|last1=Anderton|first1=Ethan|title=Sylvester Stallone Stars in 'Crash' Style Drama 'Reach Me' Trailer|url=http://www.firstshowing.net/2014/sylvester-stallone-stars-in-crash-style-drama-reach-me-trailer/|accessdate=July 7, 2014|publisher=firstshowing.net|date=July 7, 2014}}&lt;/ref&gt; and the second official trailer was released on September 25, 2014.&lt;ref&gt;{{cite web | last=Movieclips Trailers | date=September 25, 2014 | title=Reach Me Official Trailer #2 (2014) - Sylvester Stallone, Nelly Movie HD | work=[[YouTube]] | url=https://www.youtube.com/watch?v=bwvCSfBQekI | access-date=December 29, 2017}}&lt;/ref&gt;

2564 Pai went on to appear in several [[feature film]]s and television shows including the [[Burt Reynolds]] [[action film]] ''[[Sharky's Machine (film)|Sharky's Machine]]'' (1981),&lt;ref name=Fame82&gt;{{cite news |url=https://www.newspapers.com/newspage/294833047/ |title=Fame |date=February 19, 1982 |newspaper=The Marion Star |access-date=13 September 2022 |quote=In Burt Reynolds' action flick, &quot;Sharky's Machine,&quot; several high-class call girls fall victim to a terrorist. One of those girls is played by alluring 19-year-old actress-model Suzee Pai. Of Chinese-French stock, the lovely Ms. Pai has also graced the pages of Penthouse magazine.}}&lt;/ref&gt; the [[Sylvester Stallone]] action film ''[[First Blood]]'' (1982),{{efn|The scene with Pai was deleted, but is a bonus feature on the &quot;Ultimate Edition&quot; DVD}} [[John Carpenter]]'s ''[[Big Trouble in Little China]]'' (1986), ''Jakarta'' (1988),{{efn|She is credited as '''Sue Francis Pai'''; the film had an Indonesian release, {{ill|Jakarta (1988 film)|id|Peluru dan Wanita}}}} and ''[[The Cosby Show]]'' (1984). She had a reoccurring role in the [[NBC]] [[comedy-drama]] ''[[Tattingers]]'' (1988-1989).

2565 * [[Sylvester Stallone]]

2566 '''''Ratchet &amp; Clank''''' is a 2016 animated [[science fiction film|science fiction]] [[comedy film]] produced by [[Rainmaker Entertainment]] and distributed by [[Gramercy Pictures]]. Based on [[Insomniac Games]]' [[Ratchet &amp; Clank|video game series of the same name]], the film was directed by [[Kevin Munroe]] and co-directed by Jericca Cleland. [[James Arnold Taylor]] and [[David Kaye (voice actor)|David Kaye]] reprise their roles as the titular characters from the video games, alongside [[Jim Ward (voice actor)|Jim Ward]] and [[Armin Shimerman]] as their respective characters. The film also stars the voices of [[Paul Giamatti]], [[John Goodman]], [[Bella Thorne]], [[Rosario Dawson]], [[Vincent Tong (voice actor)|Vincent Tong]], Andrew Cownden, and [[Sylvester Stallone]].

2567 * [[Sylvester Stallone]] as Lieutenant Victor Von Ion: Drek's robotic lieutenant.&lt;ref name=&quot;Cast Update&quot;/&gt; He is an original character created for the film.

2568 The second single, &quot;[[I Hope You Find It#Cher version|I Hope You Find It]]&quot;, premiered on September 23, 2013, during Cher's ''[[Today (U.S. TV program)|Today Show]]'' concert, where she also performed &quot;Woman's World&quot; and &quot;[[Believe (Cher song)|Believe]]&quot;. It was further performed on the ''[[Late Show with David Letterman]]'' on September 24, 2013, on ''[[Live! with Kelly and Michael]]'' on October 1, 2013,&lt;ref&gt;[http://www.antimusic.com/news/13/September/ts20Cher_To_Perform_On_LIVE_with_Kelly_and_Michael.shtml#.Uj7icj-1M_w Cher To Perform On LIVE with Kelly and Michael] {{Webarchive|url=https://web.archive.org/web/20210519073719/https://www.antimusic.com/news/13/September/ts20Cher_To_Perform_On_LIVE_with_Kelly_and_Michael.shtml#.Uj7icj-1M_w | date=May 19, 2021 }}, Antimusic.com&lt;/ref&gt; and on German TV show ''[[Wetten, dass..?]]'' on October 5, 2013.&lt;ref&gt;[http://www.promicabana.de/wetten-dass-bremen-cher-harrison-ford-sylvester-stallone/ {{&quot;'}}Wetten, dass..?' in Bremen mit Cher, Harrison Ford &amp; Sylvester Stallone&quot;] {{Webarchive|url=https://web.archive.org/web/20210519073718/https://www.promicabana.de/wetten-dass-bremen-cher-harrison-ford-sylvester-stallone/ |date=May 19, 2021 }}, Promicabana.de. September 22, 2013&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://www.amazon.de/Hope-Find-Womans-World-2track/dp/B00F4IEAXQ/ref=sr_1_1?s=music&amp;ie=UTF8&amp;qid=1379853466&amp;sr=1-1&amp;keywords=cher+i+hope+you+find+it |title=I Hope You Find It/Woman's World (2track) |website=Amazon.com}}&lt;/ref&gt; She then continued promotion of the single in the UK, performing on [[The X Factor (UK TV series)|''The X Factor'']] on October 13, 2013,&lt;ref&gt;[http://www.digitalspy.co.uk/tv/s103/the-x-factor/news/a520587/cher-ellie-goulding-for-first-x-factor-live-results-show.html &quot;Cher, Ellie Goulding for first 'X Factor' live results show&quot;] {{Webarchive|url=https://web.archive.org/web/20150924234834/http://www.digitalspy.co.uk/tv/s103/the-x-factor/news/a520587/cher-ellie-goulding-for-first-x-factor-live-results-show.html |date=September 24, 2015 }}, Digital Spy. October 3, 2013.&lt;/ref&gt; on ''[[The Graham Norton Show]]'' on October 18, 2013, and then on ''[[Vivement dimanche!]]'' in [[Paris]] on November 24, 2013. An official lyric video for the single was released on September 24, 2013.&lt;ref&gt;[https://www.youtube.com/watch?v=0FBFVLunYA4 Cher - I Hope You Find It [OFFICIAL LYRIC VIDEO&lt;nowiki&gt;]&lt;/nowiki&gt;] on [[YouTube]]. {{Webarchive|url=https://web.archive.org/web/20210524152455/https://www.youtube.com/watch?v=0FBFVLunYA4 |date=May 24, 2021 }}&lt;/ref&gt; In 2014 &quot;I Hope You Find It&quot; debuted on ''Billboard'''s [[Adult Contemporary (chart)|Adult Contemporary]] chart at number 24, becoming Cher's 31st entry on that chart during the course of five decades.&lt;ref&gt;{{cite magazine|title=Chart Highlights: American Authors Are 'Best' On Adult Pop Songs|url=http://www.billboard.com/articles/columns/chart-beat/5923089/chart-highlights-american-authors-are-best-on-adult-pop-songs|magazine=Billboard|access-date=March 3, 2014|archive-date=March 6, 2014|archive-url=https://web.archive.org/web/20140306042427/http://www.billboard.com/articles/columns/chart-beat/5923089/chart-highlights-american-authors-are-best-on-adult-pop-songs|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=Cher Scores 31st Billboard Adult Contemporary Hit With 'I Hope You Find It'|date=March 4, 2014|url=http://chernews.blogspot.com/2014/03/cher-scores-29th-billboard-adult.html?spref=tw|publisher=Cher News|access-date=March 7, 2014|archive-date=March 25, 2014|archive-url=https://web.archive.org/web/20140325022702/http://chernews.blogspot.com/2014/03/cher-scores-29th-billboard-adult.html?spref=tw|url-status=live}}&lt;/ref&gt;

2569 * ''[[Creed (film)|Creed]]'', 2015 film starring [[Michael B. Jordan]] and [[Sylvester Stallone]]

2570 * ''[[Creed II]]'', 2018 film starring [[Michael B. Jordan]] and [[Sylvester Stallone]]

2571 *June 14 - [[Gerry Cooney]] &amp; [[Sylvester Stallone]]

2572 *[[Sylvester Stallone]]

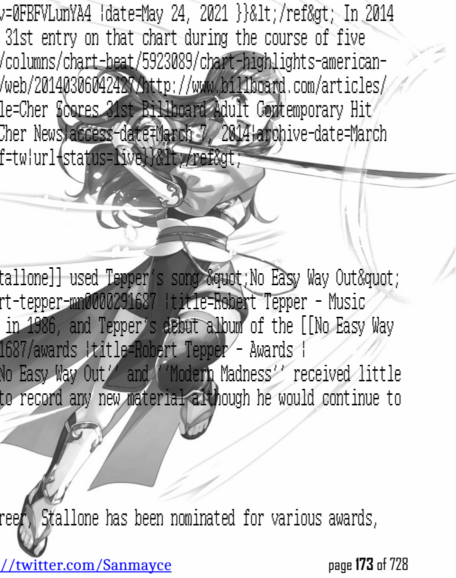
2573 After signing with Scotti Brothers and moving to Los Angeles in 1985, Tepper found himself in the public eye after actor/director [[Sylvester Stallone]] used Tepper's song &quot;No Easy Way Out&quot; in the movie ''[[Rocky IV]]''.&lt;ref name=&quot;allmusic1&quot;&gt;{{cite web|last=Demalon |first=Tom |url=http://www.allmusic.com/artist/robert-tepper-mn0000291687 |title=Robert Tepper - Music Biography, Credits and Discography |publisher=AllMusic |access-date=2013-06-14}}&lt;/ref&gt; The track climbed to No. 22 on Billboard's Hot 100 in 1986, and Tepper's debut album of the [[No Easy Way Out (album)|same name]] peaked at No. 144.&lt;ref&gt;{{cite web|author=Robert Tepper |url=http://www.allmusic.com/artist/robert-tepper-mn0000291687/awards |title=Robert Tepper - Awards | publisher=AllMusic |access-date=2013-06-14}}&lt;/ref&gt; However, the 1988 follow-up album ''Modern Madness'' was a commercial failure. Both ''No Easy Way Out'' and ''Modern Madness'' received little promotion from the label [[Scotti Bros. Records]]. As a result, Tepper separated from the label in protracted legal proceedings and was unable to record any new material although he would continue to write. A third album ''No Rest for the Wounded Heart'' would follow eight years later in 1996.&lt;ref name=&quot;allmusic1&quot;/&gt;

2574 <title>List of awards and nominations received by Sylvester Stallone</title>

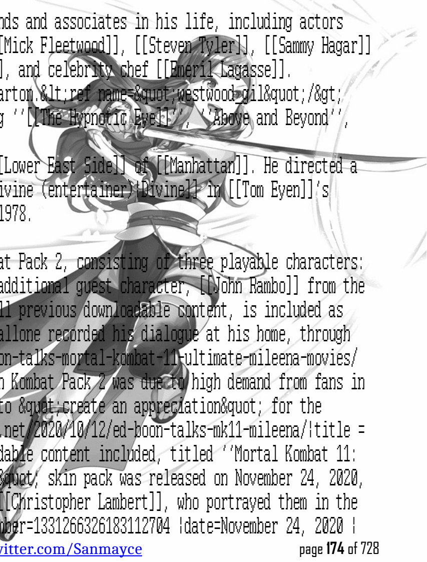
2575 ! colspan=&quot;2&quot; style=&quot;text-align:center;&quot; | '''[[Sylvester Stallone]] awards'''

2576 | colspan=3 style=&quot;text-align:center;&quot; | [[File:Sylvester Stallone Cannes 2014 2.jpg|200px]]

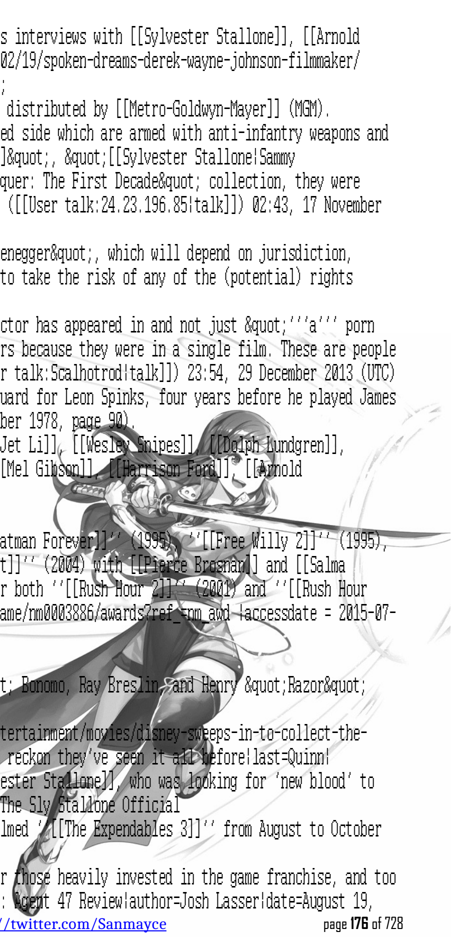
2577 [[Sylvester Stallone]] is an American actor, [[screenwriter]], and [[film director]] who has appeared in multiple film roles. Throughout his career, Stallone has been nominated for various awards, including [[Academy Award]]s, [[British Academy Film Awards]], [[Golden Raspberry Awards]], and [[People's Choice Awards]].



```
2578 *{{IMDb name¦id=0000230¦name=Sylvester Stallone¦section=awards}}
2579 {{Sylvester Stallone}}
2580 [[Category:Sylvester Stallone¦Awards]]
2581  ¦ShortSummary=Don visits the [[Rocky Balboa]] statue at the [[Philadelphia Museum of Art]], uncovering the real story behind the character through boxer [[Chuck Wepner]], who inspired [[Sylvester Stallone]] to write ''[[Rocky]]''; examines the tombstones of a murdered dentist and his mistress, [[Emma Cunningham]], who are eternally linked in [[Brooklyn]]'s [[Green-Wood Cemetery]]; investigates [[San Diego]]'s &quot;[[Unconditional Surrender (sculpture)¦Unconditional Surrender]]&quot; sculpture of the [[Alfred Eisenstaedt]] photo of a [[World War II]] sailor kissing a nurse; explores [[Los Angeles¦L.A.]]'s [[Elizabeth Lake (Los Angeles County, California)¦Elizabeth Lake]] (&quot;Devils Lake&quot;) in [[Angeles National Forest]], home of the [[Thunderbird (cryptozoology)¦Thunderbird]], a 90-foot winged creature; discovers [[Hilo, Hawaii]]'s town clock's hands are permanently frozen at 1:04 a.m. when a [[tsunami]] hit on May 22, 1960; and learns of the [[Nez Perce people¦Nez Perce]] woman [[Watkuese]], who saved [[Lewis and Clark]]'s [[Lewis and Clark Expedition¦expedition]], making it possible for the &quot;Captain's Return&quot; statue at [[St. Louis Arch]]'s base.
2582 ¦Guests=Sylvester Stallone, Steve Wiebe
2583 ¦ [[Golden Raspberry Award for Worst Screenplay¦Worst Screenplay]]{{efn¦shared with [[Kevin Jarre]] and [[Sylvester Stallone]]}}
2584 [[San Francisco]]'s [[Dead Kennedys]] made a career out of mentioning Reagan in songs like &quot;Moral Majority&quot;, &quot;[[We've Got a Bigger Problem Now]],&quot; &quot;[[Bleed for Me (Dead Kennedys song)¦Bleed for Me]]&quot;, and the track &quot;Kinky Sex Makes the World Go Round&quot;, a spoken-word piece about [[World War III]] formatted as an [[phone sex¦erotic phone call]] between Margaret Thatcher and Reagan's fictitious [[United States Secretary of War¦Secretary of War]].&lt;ref name=&quot;Cohen and Kaufman, 2015&quot; /&gt;&lt;ref name=&quot;Bay Area Punk&quot;&gt;{{cite web¦title=Three Anti-Reagan Punk Bands From San Francisco¦url=http://bayareapunk.com/blog/anti-reagan-sf.html¦website=bayareapunk.com¦access-date=January 5, 2017}}&lt;/ref&gt; The band's 1986 studio album, ''[[Bedtime for Democracy]]'', is a play on Reagan's film ''[[Bedtime for Bonzo]]'' and features a multitude of songs about Reagan. &quot;Potshot Heard Round the World&quot; is about US military actions in the Middle East, &quot;with Reagans and [[Muammar Gaddafi¦Gaddafi]]s cast as cartoon villains and heroes.&quot; Reagan plays the title role in the song, &quot;Rambozo the Clown&quot;, a [[portmanteau]] of [[Sylvester Stallone]]'s ''[[Rambo (film series)¦Rambo]]'' franchise and [[Bozo the Clown]] from children's [[daytime TV]].&lt;ref name=&quot;Cohen and Kaufman, 2015&quot; /&gt; The Dead Kennedys were done in by a lawsuit against their inclusion of [[H. R. Giger]]'s ''[[Penis Landscape]]'' painting as an insert for the album ''[[Frankenchrist]]''. Singer [[Jello Biafra]] was attracted to Giger's work as soon as he saw it, saying, &quot;This picture is like Reagan America on parade.&quot;&lt;ref name=&quot;Silverberg, 2014&quot;&gt;{{cite news¦last1=Silverberg¦first1=Michael¦title=The obscenity trial that made H. R. Giger an icon for punk rock and free speech¦url=https://qz.com/210900/the-obscenity-trial-that-made-h-r-giger-an-icon-for-punk-rock-and-free-speech/¦access-date=August 26, 2017¦work=Quartz¦date=May 20, 2014}}&lt;/ref&gt;
2585 ¦Author = Sylvester Stallone
2586 The school later purchased the costume of a ram for a new mascot, which was named &quot;Rambo&quot; after the protagonist [[John Rambo]] in the 1980s film series starring [[Sylvester Stallone]]. By 2001, the costume had suffered enough wear that a new Rambo costume was purchased.
2587 She has also interviewed [[Academy Award]] winners such as [[Paul Newman]], [[Elizabeth Taylor]], [[Jack Lemmon]], [[Dennis Hopper]], [[Robert Altman]], [[Steven Spielberg]],&lt;ref&gt;[http://archiviostorico.corriere.it/2004/marzo/26/Spielberg_mia_lista_dei_400_co_9_040326085.shtml &quot;La mia lista dei 400 italiani&quot;] Corriere della Sera - March 26, 2004&lt;/ref&gt; [[Woody Allen]], [[Francis Ford Coppola]], [[Martin Scorsese]], the [[Coen brothers]], [[Oliver Stone]], [[Spike Lee]], [[Meryl Streep]],&lt;ref&gt;Meryl Streep: &quot;E' la mia Africa&quot; - Corriere della Sera - February 22, 1986&lt;/ref&gt; [[Robert De Niro]], [[Daniel Day-Lewis]], [[Robin Williams]], [[Al Pacino]], [[Kirk Douglas]], [[Brad Pitt]], [[Demi Moore]], [[Dustin Hoffman]], [[Jodie Foster]], [[Robert Redford]], [[Jane Fonda]], [[Julia Roberts]], [[Richard Gere]], [[Susan Sarandon]], [[Sylvester Stallone]], [[Sarah Jessica Parker]], [[John Turturro]], [[Federico Fellini]], [[Marcello Mastroianni]], [[Franco Zeffirelli]], [[Bernardo Bertolucci]], [[Sophia Loren]], [[Roberto Benigni]] and international music stars including [[Frank Sinatra]], [[Sammy Davis Jr]]., [[Luciano Pavarotti]], [[Ella Fitzgerald]], [[Aretha Franklin]], [[Joan Baez]],&lt;ref&gt;[http://archiviostorico.corriere.it/2007/febbraio/17/Joan_Baez_PROTESTO_ANCORA_co_9_070217037.shtml Joan Baez: &quot;Protesto ancora&quot;] Corriere della Sera - February 17, 2007&lt;/ref&gt; [[Plácido Domingo]], [[Liza Minnelli]],&lt;ref&gt;Liza Minnelli: &quot;Così sono ritornata star&quot; Corriere della Sera - September 29, 1987&lt;/ref&gt; [[Stevie Wonder]], [[Bruce Springsteen]], [[Mick Jagger]], [[Prince (musician)¦Prince]] and [[Patti Smith]].
2588 ¦ style=&quot;text-align:center;&quot; ¦ &lt;ref&gt;{{cite web¦first=Justin¦last=Kroll¦title='The Expendables 3' Premiere: 'Break a Leg,' Says Sylvester Stallone to Harrison Ford¦url=https://variety.com/2014/scene/news/harrison-ford-sylvester-stallone-the-expendables-3-premiere-1201281124/¦work=[[Variety (magazine)¦Variety]]¦access-date=November 10, 2014¦date=August 12, 2014}}&lt;/ref&gt;
2589 A severe alcoholic also known simply as ''Kah'', who has a strong disdain for foreigners, particularly the British. He is portrayed as a violent drunk, who often verbally, and occasionally physically, assaults other Freedom Town residents. [[Sylvester Stallone]] voiced Dunkaholic in most of the episodes, and he has been quoted as saying that role was the &quot;most demeaning&quot; of his career.&lt;ref&gt;http://www.gq.com/entertainment/celebrities/201009/sylvester-stallone-yo-michael-hainey-cop-land-rocky-rambo&lt;/ref&gt;
2590 Dimitri Vegas is a big comic book fan and owns a large collection, with his favourite being the [[Spider-Man]] series.&lt;ref&gt;{{Cite web¦url=http://www.bbc.co.uk/newsbeat/article/33346777/dimitri-vegas-edm-is-music-of-this-generation¦title=Dimitri Vegas: EDM is music of this generation¦website=BBC¦date=7 October 2015}}&lt;/ref&gt; In 2017, Dimitri married Belgian-born DJ and producer [[Mattn]] in Ibiza.&lt;ref&gt;{{cite web¦author=Provost, Brittany¦url=https://www.edmtunes.com/2017/09/dimitri-vegas-has-tied-the-knot-stunning-doesnt-even-describe-the-pictures/ ¦title=Dimitri Vegas Has Tied the Knot With MATTN, And Stunning Doesn't Even Begin To Describe Their Photos¦website=Edmtunes.com¦date=2017-08-11¦access-date=2019-09-13}}&lt;/ref&gt; In 2021 the couple have a son, London Thivaios. Vegas has also appeared in several feature films that include ''[[Jurassic World Dominion¦Jurassic World: Dominion]]'', ''[[The Bouncer (film)¦The Bouncer]]'', ''[[Men in Black: International]]'', ''[[Rambo: Last Blood]], [[Yummy (film)¦Yummy]],'' ''[[H4Z4RD]]'' and ''[[It's a Wonderful Knife]]''.&lt;ref&gt;{{Cite web¦last=Halperin¦first=Shirley¦date=2019-05-19¦title=Dimitri Vegas Talks 'Rambo V: Last Blood' Role, Working With Sylvester Stallone¦url=https://variety.com/2019/music/news/dimitri-vegas-rambo-v-last-blood-sylvester-stallone-1203219408/¦access-date=2020-07-30¦website=Variety¦language=en}}&lt;/ref&gt;
2591 ¦ ''[[The Expendables 3]]'' ¦¦ [[Lionsgate Films¦Lionsgate]] / [[Alchemy (company)¦Millennium Films]] / [[Nu Image]] ¦¦ [[Patrick Hughes (director)¦Patrick Hughes]] (director); [[Sylvester Stallone]], Creighton Rothenberger, Katrin Benedikt (screenplay); Sylvester Stallone, [[Jason Statham]], [[Antonio Banderas]], [[Jet Li]], [[Wesley Snipes]], [[Dolph Lundgren]], [[Kelsey Grammer]], [[Randy Couture]], [[Terry Crews]], [[Kellan Lutz]], [[Ronda Rousey]], [[Glen Powell]], [[Victor Ortiz]], [[Robert Davi]], [[Mel Gibson]], [[Harrison Ford]], [[Arnold Schwarzenegger]] ¦¦ style=&quot;text-align:center;&quot;¦&lt;ref&gt;{{cite web¦date=2014-05-19¦title=The Expendables 3 Release¦url=https://www.comingsoon.net/news/movienews.php?id=118505¦access-date=2014-11-03¦publisher=Comingsoon.net}}&lt;/ref&gt;
2592 ¦  align = &quot;center&quot; ¦ 2019 ¦¦ ''[[Rambo: Last Blood]]'' ¦¦ Víctor Martínez ¦¦ ¦¦ align = &quot;center&quot; ¦ &lt;ref name=&quot;deadlineoct23&quot;&gt;{{Cite web¦url=https://deadline.com/2018/10/sylvester-stallone-rambo-last-blood-oscar-jaenada-1202488090/¦title=Sylvester Stallone's 'Rambo: Last Blood' Adds Oscar Jaenada¦access-date=2 April 2019¦last=N'Duka¦first=Amanda¦date=23 October 2018¦website=Deadline}}&lt;/ref&gt;
2593 Bergvall is the [[Academy Awards¦Academy Award]] nominated [[filmmaker]] behind some of the biggest brand stories of the past decades. With a unique blend of Hollywood and brand experience, Bergvall is known for the short film ''Victor'' (1999), the [[thriller (genre)¦thriller]] ''[[Den osynlige]]'' (2002, remade as ''[[The Invisible (2007 film)¦The Invisible]]'' in 2007) and [[screenplay]] adaption of ''[[Tunnels (novel)¦Tunnels]]'', the [[United Kingdom¦British]] [[novel]] series by [[Roderick Gordon]] and Brian Williams, as well as some of the biggest and most shareable branded content short films of the past decade, working with brands like Google, AT&amp;T, Ford, MARS, Intuit, and many more, as well as celebrities like [[Sylvester Stallone]], [[Cristiano Ronaldo]], [[Dua Lipa]], [[John Cena]], [[Ed Sheeran]] and many more.
2594 With the rise of digital media, Bergvall specialized in highly shareable branded content. As Partner and Chief Creative Officer with Shareability,&lt;ref&gt;{{Cite web ¦title=S h a r e a b i l i t y ¦url=http://www.shareability.com ¦access-date=2017-01-13 ¦website=Shareability}}&lt;/ref&gt; Bergvall produced some of the biggest hits in the world over the last few years, working with celebrities like [[Sylvester Stallone]], [[Cristiano Ronaldo]], [[Tim McGraw]], [[Dua Lipa]], [[John Cena]], [[Ed Sheeran]] and many more.&lt;ref&gt;{{Cite web ¦title=Joel Bergvall - Biography ¦url=https://www.imdb.com/name/nm0075150/bio/ ¦access-date=2024-03-21 ¦website=IMDb ¦language=en-US}}&lt;/ref&gt;
2595 The film's narrative is driven primarily by Shep Gordon as he recounts the story of his career. Supporting interviews feature some of the many friends and associates in his life, including actors [[Michael Douglas]], [[Sylvester Stallone]], [[Tom Arnold (actor)¦Tom Arnold]] and [[Mike Myers]], musicians [[Alice Cooper]], [[Willie Nelson]], [[Mick Fleetwood]], [[Steven Tyler]], [[Sammy Hagar]] and [[Anne Murray]], film producer [[Carolyn Pfeiffer]], record producer [[Bob Ezrin]], [[National Basketball Association¦NBA]] coach [[Don Nelson]], and celebrity chef [[Emeril Lagasse]].
2596 From his Glendale offices near Hollywood, Gil worked with many actors throughout his career, including Sylvester Stallone, Lily Tomlin, and Dolly Parton.&lt;ref name=&quot;westwood_gil&quot;/&gt; Such work often brought international media attention. He was also a technical director or advisor on several films and television series, including ''[[The Hypnotic Eye]]'', ''Above and Beyond'', and ''[[The Eleventh Hour (1962 TV series)¦The Eleventh Hour]]''.&lt;ref name=&quot;gbo&quot;/&gt;
2597 Link directed [[Off-Off-Broadway¦off-off-Broadway]] theatre, working primarily at [[Caffe Cino]] and [[La MaMa Experimental Theatre Club]] in the [[Lower East Side]] of [[Manhattan]]. He directed a young [[Robert De Niro]] in ''Glamour, Glory and Gold'' and a young [[Sylvester Stallone]] in [[Somerset Maugham]]'s ''Rain''. He also directed [[Divine (entertainer)¦Divine]] in [[Tom Eyen]]'s ''[[Women Behind Bars]]'' at La MaMa and at the [[Astor Place Theater]] in 1974, and in ''[[The Neon Woman]]'' at [[Hurrah (nightclub)¦Hurrah]] in 1978.
2598 ¦ align=&quot;center&quot; ¦ [[Steve Agee]] (motion-capture)&lt;br&gt;[[Sylvester Stallone]] (voice)
2599 On October 6, 2020, a teaser trailer for an additional DLC pack was uploaded, promising more information on October 8. This was revealed to be Kombat Pack 2, consisting of three playable characters: [[Mileena]], who was last playable in ''Mortal Kombat X''; [[Rain (Mortal Kombat)¦Rain]], who was last playable in the 2011 reboot via DLC; and an additional guest character, [[John Rambo]] from the [[Rambo (franchise)¦''Rambo'' franchise]], with original actor [[Sylvester Stallone]] providing his voice and likeness. Kombat Pack 2, along with all previous downloadable content, is included as part of the game's enhanced ''Ultimate'' re-release.&lt;ref name=&quot;blog.playstation.com&quot;/&gt; To work around the [[COVID-19 pandemic]], Stallone recorded his dialogue at his home, through [[Zoom (software)¦Zoom]] conference calls with the Netherrealm staff.&lt;ref&gt;{{Cite web¦url=https://www.mortalkombatonline.com/t/mk11/nycc-ed-boon-talks-mortal-kombat-11-ultimate-mileena-movies/0QLgOILq4A8e¦title = NYCC: Ed Boon Talks Mortal Kombat 11 Ultimate, Mileena &amp; Movies¦date = 14 October 2020}}&lt;/ref&gt; Mileena's inclusion in Kombat Pack 2 was due to high demand from fans in the wake of her absence in the game previously. Ed Boon revealed that the motivation for her exclusion from the base game was that the team wanted to &quot;create an appreciation&quot; for the character, after getting constant replies from fans on social media demanding her return.&lt;ref&gt;{{Cite web¦url=https://www.playstationlifestyle.net/2020/10/12/ed-boon-talks-mk11-mileena/¦title = MK11 Mileena Was Delayed in Order to 'Create an Appreciation'¦date = 12 October 2020}}&lt;/ref&gt; An enhanced version of the game with all downloadable content included, titled ''Mortal Kombat 11: Ultimate'', was released on November 17, 2020 for the PlayStation 4, [[PlayStation 5]], Xbox One, and [[Xbox Series X/S]]. A &quot;Klassic MK Movie&quot; skin pack was released on November 24, 2020, adding new skins for Johnny Cage, Sonya Blade and Raiden using the respective voices and likenesses of [[Linden Ashby]], [[Bridgette Wilson]], and [[Christopher Lambert]], who portrayed them in the 1995 ''Mortal Kombat'' film.&lt;ref&gt;{{cite tweet ¦author=Mortal Kombat 11 Ultimate ¦author-link=Mortal Kombat 11 Ultimate ¦user=MortalKombat ¦number=1331266326183112704 ¦date=November 24, 2020 ¦
```



```
title=Look familiar? The Klassic MK Movie Skin pack with voices and likenesses from the og stars is available for separate purchase today! #MKUltimate https://t.co/38NxytBWm9 |language=en |access-date=July 27, 2021 |archive-url=https://web.archive.org/web/20210702180103/https://twitter.com/MortalKombat/status/1331266326183112704 |archive-date=July 2, 2021 |url-status=live}}&lt;/ref&gt;
2600 |'''''[[Oscar (1991 film)|Oscar]]'''''||[[John Landis]]||[[Sylvester Stallone]], [[Ornella Muti]], [[Peter Riegert]]||United States||&lt;ref&gt;{{cite web|url=http://allmovie.com/work/oscar-36689|work=Allmovie|accessdate=March 2, 2010|title=Oscar: Overview: Allmovie|author=Deming, Mark}}&lt;/ref&gt;
2601 |'''''[[Cop Land]]'''''||[[James Mangold]]||[[Sylvester Stallone]], [[Harvey Keitel]], [[Ray Liotta]]||United States||&lt;ref&gt;{{cite web|url=http://allmovie.com/work/cop-land-157716|accessdate=March 1, 2010|title=Cop Land: Overview: Allmovie|author=Ankeny, Jason}}&lt;/ref&gt;
2602 |'''''[[Get Carter (2000 film)|Get Carter]]'''''||[[Stephen Kay]]||[[Sylvester Stallone]], [[Miranda Richardson]]||United States||Crime thriller&lt;ref&gt;{{cite web|url=http://www.allmovie.com/work/get-carter-214069|work=Allmovie|accessdate=February 18, 2010|author=Deming, Mark|title=Get Carter: Overview: Allmovie}}&lt;/ref&gt;
2603 While living in [[New York City|New York]], Balfe played the minor role of an employee of the magazine ''Runway'' in the 2006 film ''[[The Devil Wears Prada (film)|The Devil Wears Prada]]''.&lt;ref name=&quot;themodelscouts&quot;/&gt; In 2009, after a decade-long modelling career, Balfe returned to her initial career choice and moved from New York to [[Los Angeles]],&lt;ref name=irishindependent2/&gt; spending her first year and a half in the city exclusively taking acting classes, first at the Warner Loughlin&lt;ref name=&quot;Warner Loughlin&quot;&gt;{{cite web |title=Warner Loughlin |url=https://www.imdb.com/name/nm0521762/ |website=IMDb |language=en-US |access-date=14 January 2022}}&lt;/ref&gt; Studios and then at the [[Sanford Meisner|Sanford Meisner Center]] and the Judith Weston&lt;ref name=&quot;Judith Weston&quot;&gt;{{cite web |title=Judith Weston |url=https://www.imdb.com/name/nm0922981/ |website=IMDb |access-date=14 January 2022}}&lt;/ref&gt; Studios.&lt;ref name=&quot;backstage.com2&quot;&gt;{{cite news|url=http://www.backstage.com/interview/gorgeous-determination-caitriona-balfe/|title=The Gorgeous Determination of Caitriona Balfe|date=2 April 2015|website=Backstage}}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=http://www.sunherald.com/2015/03/30/6150585_outlander-star-had-to-unlearn.html|title='Outlander' star had to unlearn her modeling skills|date=2 April 2015|archive-url=https://web.archive.org/web/20150402014313/http://www.sunherald.com/2015/03/30/6150585_outlander-star-had-to-unlearn.html|archive-date=2 April 2015|url-status=dead|df=dmy-all}}&lt;/ref&gt; Balfe has appeared in the films ''[[Super 8 (2011 film)|Super 8]]'', as the protagonist's mother, ''[[Now You See Me (film)|Now You See Me]]'', as [[Michael Caine]]'s character's wife, and ''[[Escape Plan (film)|Escape Plan]]'', as the CIA lawyer that hires [[Sylvester Stallone]]'s character.&lt;ref name=irishindependent2/&gt;&lt;ref&gt;{{Rotten Tomatoes|caitriona_balfe}}&lt;/ref&gt;
2604 Bradley visited Italy in 1987 to celebrate the 25th anniversary of Folkstudio, and he soon decided to take up permanent residence there. After returning to Italy, Bradley made a few more film appearances, mostly in Italian films but also a small role in the movie ''[[Daylight (1996 film)|Daylight]]'', starring Sylvester Stallone.&lt;ref&gt;[https://www.imdb.com/title/tt0116040/?ref_=nm_flmg_act_6 Daylight (1996)]&lt;/ref&gt;  He also appeared on six albums of spiritual- and folk-inspired music.&lt;ref name=&quot;thewildeye.co.uk&quot;/&gt;
2605 Liguori was also the first woman to host and produce her own weekly, prime-time series on The Golf Channel, &quot;Conversations with Ann Liguori,&quot; where she interviewed top names in entertainment, music, sports and business while golfing with them.&lt;ref&gt;{{cite web|url=http://www.sportsideo.com/blog/donnamartini91/all/golf-with-wfan-radios-golf-correspondent-ann-liguori.html |archive-url=https://web.archive.org/web/20130927194815/http://sportsideo.com/blog/donnamartini91/all/golf-with-wfan-radios-golf-correspondent-ann-liguori.html |url-status=dead |archive-date=2013-09-27 |title=Golf with WFAN Radio's golf correspondent Ann Liguori! &amp;#124; Sportsideo.com member Donna Martini |publisher=Sportsideo.com |date=April 7, 2013 |accessdate=2013-10-18 }}&lt;/ref&gt; The series aired on TGC the first five years of the Channel's inception. Guests included an 'A List' of celebrities that Liguori booked for the show including [[Kevin Costner]], [[Sylvester Stallone]], [[Celine Dion]], [[Vince Gill]], [[Smokey Robinson]], [[Joe Pesci]], [[Matthew McConaughey]], among others. The interviews focus on the celebrity's success stories and their passion for golf. She turned those interviews into her book, &quot;A Passion for Golf, Celebrity Musings About the Game&quot; which came out in a second printing in 2007.&lt;ref&gt;{{cite book|author=Ann Ligouri (Author) |title=A Passion for Golf: Celebrity Musings About the Game: Ann Ligouri: 9780878339723: Amazon.com: Books |date=October 1, 1997 |isbn=0878339728 }}&lt;/ref&gt;
2606 After supporting roles in ''[[Shocker (film)|Shocker]]'', ''[[Against All Odds (1984 film)|Against All Odds]]'', and ''[[The Karate Kid]]'', Sam played arm wrestler Harry Boscoe in the 1987 [[Sylvester Stallone]] film ''[[Over the Top (1987 film)|Over The Top]]''.  He has since appeared in small roles in many television shows and movies.
2607 | book        = [[Thomas Meehan (writer)|Thomas Meehan]]&lt;br&gt;[[Sylvester Stallone]]
2608 | basis       =''[[Rocky]]''&lt;br&gt;by Sylvester Stallone
2609 '''''Rocky the Musical''''' (originally '''''Rocky: Das Musical''''')&lt;ref name=&quot;knocks&quot;&gt;{{cite web | url=https://www.theguardian.com/stage/2012/nov/20/rocky-das-musical-hamburg | title=Rocky musical knocks out critics in Hamburg | work=[[The Guardian]] | date=20 November 2012 | access-date=27 December 2013}}&lt;/ref&gt;&lt;ref&gt;{{cite web | url=http://news.sky.com/story/1012477/rocky-comes-out-fighting-with-das-musical | title=Rocky Comes Out Fighting With 'Das Musical' | publisher=[[Sky News]] | work=news.sky.com | date=16 November 2012 | access-date=27 December 2013}}&lt;/ref&gt;  is a 2012 [[Musical theatre|musical]] with music by [[Stephen Flaherty]], lyrics by [[Lynn Ahrens]], and a book by [[Thomas Meehan (writer)|Thomas Meehan]] and [[Sylvester Stallone]], based on the [[Rocky|1976 film of the same name]] written by Stallone. The show held its world premiere in [[Hamburg]] in 2012 and opened on [[Broadway theatre|Broadway]] in 2014 at the [[Winter Garden Theatre]].
2610 The musical is based on the 1976 film ''[[Rocky]]'', with a screenplay by [[Sylvester Stallone]].&lt;ref&gt;{{cite web | url=https://www.usatoday.com/story/life/people/2013/04/28/rocky-musical-headed-to-broadway/2119697/ | title='Rocky' musical bound for Broadway in 2014 | publisher=[[USA Today]] | work=usatoday.com | date=28 April 2013 | access-date=27 December 2013}}&lt;/ref&gt; The film itself was made on a budget of $1,075,000,&lt;ref&gt;{{cite web | url=http://www.shortlist.com/entertainment/films/15-things-you-(probably)-didn't-know-about-rocky#item-7 | title=15 THINGS YOU (PROBABLY) DIDN'T KNOW ABOUT ROCKY | publisher=Shortlist | work=shortlist.com | access-date=28 December 2013}}&lt;/ref&gt; shot in 28 days&lt;ref&gt;{{cite web | url=http://www.broadwayworld.com/article/Official-ROCKY-to-Open-at-Winter-Garden-Theatre-on-313-Previews-Begin-211-20130826 | title=ROCKY to Open at Winter Garden Theatre on 3/13; Previews Begin 2/11 | publisher=Broadway World | work=broadwayworld.com | date=26 August 2013 | access-date=28 December 2013}}&lt;/ref&gt; and was a [[sleeper hit]],&lt;ref&gt;{{cite web | url=https://ew.com/article/2002/02/19/how-rocky-nabbed-best-picture/ | title=The Right Hook | work=[[Entertainment Weekly]] | date=19 February 2002 | access-date=28 December 2013}}&lt;/ref&gt; earning $225 million in global box office receipts&lt;ref&gt;{{cite web | url=http://www.huffingtonpost.co.uk/2013/06/18/low-budget-movies-that-made-millions_n_3459590.html#slide=2582768 | title=14 Low Budget Movies That Made Millions | publisher=[[The Huffington Post]] | work=huffingtonpost.co.uk | date=26 June 2013 | access-date=28 December 2013}}&lt;/ref&gt; becoming the [[1976 in film|highest-grossing film of 1976]]&lt;ref&gt;{{cite web | url=http://news.msn.com/offbeat/yo-adrian-rocky-devotees-set-to-make-tribute-run | title=Yo, Adrian! Rocky devotees set to make tribute run | publisher=[[MSN]] | date=12 May 2013 | access-date=28 December 2013 | archive-url=https://web.archive.org/web/20131230234731/http://news.msn.com/offbeat/yo-adrian-rocky-devotees-set-to-make-tribute-run | archive-date=30 December 2013 | url-status=dead }}&lt;/ref&gt; and went on to win three [[Academy Award|Oscars]],&lt;ref&gt;{{cite web | url=http://www.oscars.org/oscars/ceremonies/1977 | title=The 49th Academy Awards (1977) Nominees and Winners | publisher=[[Academy Award]] | work=oscars.org | access-date=28 December 2013}}&lt;/ref&gt; including [[Academy Award for Best Picture|Best Picture]].&lt;ref&gt;{{cite web | url=http://www.today.com/id/16689513/ns/today-today_entertainment/t/rockys-oscar-tko-still-echoes-stings/#.Ur8kzqWFX8s | title=Rocky's' Oscar TKO in '76 still echoes, stings | publisher=[[Today (U.S. TV program)|Today]] | work=today.com | date=19 January 2007 | access-date=28 December 2013}}&lt;/ref&gt; Having been in the works for eight years,&lt;ref&gt;{{cite web | url=http://artsbeat.blogs.nytimes.com/2011/06/01/gonna-fly-soon-a-rocky-musical-is-moving-ahead/ | title=A 'Rocky' Musical Is Moving Ahead | work=[[The New York Times]] | date=1 June 2011 | access-date=29 December 2013}}&lt;/ref&gt; a workshop was held in [[New York City]] in April 2011,&lt;ref&gt;{{cite web | url=https://huffingtonpost.com/2011/06/02/rocky-musical-is-being-wr_n_870012.html | title='Rocky' Broadway Musical Is Being Written, Hopes To Open By 2013 | publisher=Huffington Post | work=huffingtonpost.com | date=6 February 2011 | access-date=27 December 2013}}&lt;/ref&gt; with [[Andy Karl]] playing Rocky and [[Lisa Brescia]] playing Adrian.&lt;ref&gt;{{cite web | url=http://www.broadway.com/buzz/156598/yo-andy-karl-stage-vet-plays-lead-in-reading-of-broadway-bound-rocky-musical/ | title=Yo, Andy Karl! Stage Vet Plays Lead in Reading of Broadway-Bound Rocky Musical | work=broadway.com | date=1 June 2011 | access-date=29 December 2013}}&lt;/ref&gt; Following the reading it was officially confirmed in November by producer Sylvester Stallone alongside boxers and co-producers&lt;ref&gt;{{cite web | url=https://www.theguardian.com/film/2011/nov/21/rocky-musical-debut-germany | title=Rocky the Musical to debut in Germany | work=[[The Guardian]] | date=21 November 2011 | access-date=28 December 2013}}&lt;/ref&gt; [[Vitali Klitschko|Vitali]] and [[Wladimir Klitschko]],&lt;ref&gt;{{cite web | url=https://www.bbc.co.uk/news/entertainment-arts-15820805 | title=Stallone's Rocky musical to debut in Germany | publisher=[[BBC News]] | work=bbc.co.uk/news | date=21 November 2011 | access-date=28 December 2013}}&lt;/ref&gt; that the show would receive its world premiere in [[Hamburg]] in November 2012.&lt;ref&gt;{{cite web | url=https://www.telegraph.co.uk/culture/music/music-news/8904027/Rocky-The-Musical-is-coming-says-Sylvester-Stallone.html | title=Rocky The Musical is coming, says Sylvester Stallone | publisher=[[The Daily Telegraph|The Telegraph]] | work=telegraph.co.uk | date=21 November 2011 | access-date=28 December 2013}}&lt;/ref&gt;
2611 The musical, which premiere production cost around $20million&lt;ref&gt;{{cite web | url=http://artsbeat.blogs.nytimes.com/2013/04/28/rocky-musical-is-heading-to-broadway/?_r=0 | title='Rocky' Musical Is Heading to Broadway | work=[[The New York Times]] | date=28 April 2013 | access-date=27 December 2013}}&lt;/ref&gt; to produce, has a book by [[Thomas Meehan (writer)|Thomas Meehan]] and Sylvester Stallone,&lt;ref name=&quot;fight&quot;&gt;{{cite web | url=https://hollywoodreporter.com/news/rocky-musical-headed-broadway-448026 | title='ROCKY' MUSICAL HEADED TO BROADWAY | publisher=Hollywood Reporter | work=hollywoodreporter.com | date=28 April 2013 | access-date=27 December 2013}}&lt;/ref&gt; adapted from Stallone's screenplay, and is directed by [[Alex Timbers]],&lt;ref&gt;{{cite web | url=http://metro.co.uk/2011/11/23/rocky-the-musical-set-for-2012-debut-sylvester-stallone-reveals-229979/ | title=Rocky the musical set for 2012 debut, Sylvester Stallone reveals | publisher=[[Metro (British newspaper)|Metro]] | work=metro.co.uk | date=23 November 2011 | access-date=27 December 2013}}&lt;/ref&gt; with choreography by Kelly Devine, boxing choreography by [[Steven Hoggett]],&lt;ref&gt;{{cite web | url=http://www.playbill.com/news/article/156786-Rocky-Musical-Will-Get-World-Premiere-in-Fall-2012-Steven-Hoggett-Kelly-Devine-Alex-Timbers-in-the-Ring | title=Rocky Musical Will Get World Premiere in Fall 2012; Steven Hoggett, Kelly Devine, Alex Timbers in the Ring | publisher=[[Playbill]] | work=playbill.com | date=20 November 2011 | access-date=27 December 2013 | url-status=dead | archive-url=https://web.archive.org/web/20131230232655/http://www.playbill.com/news/article/156786-Rocky-Musical-Will-Get-World-Premiere-in-Fall-2012-Steven-Hoggett-Kelly-Devine-Alex-Timbers-in-the-Ring | archive-date=30 December 2013 }}&lt;/ref&gt; set design by Christopher Barreca,&lt;ref name=&quot;Creative&quot;&gt;{{cite web | url=http://www.stage-entertainment.de/musicals-shows/rocky-das-musical-hamburg/artists.html#Kreativteam | title=Creative Team | publisher=[[Stage Entertainment]] | work=stage-entertainment.de | access-date=28 December 2013}}&lt;/ref&gt; costume design by David Zinn,&lt;ref name = &quot;Creative&quot;/&gt; lighting design by [[Christopher Akerlind]],&lt;ref name = &quot;Creative&quot;/&gt; sound design by [[Peter Hylenski]],&lt;ref name = &quot;Creative&quot;/&gt; video design by [[Pablo N. Molina]]&lt;ref name = &quot;Creative&quot;/&gt; and special effects by Jeremy Chernick.&lt;ref name = &quot;Creative&quot;/&gt; A mostly original score has been penned, by [[Lynn Ahrens]] with music by [[Stephen Flaherty]].&lt;ref&gt;{{cite web | url=https://www.independent.co.uk/arts-entertainment/theatre-dance/news/sylvester-stallones-rocky-the-musical-to-transfer-to-broadway-8594516.html | title=Sylvester Stallone's Rocky: The Musical to transfer to Broadway | publisher=[[The Independent]] | work=independent.co.uk | date=29 April 2013 | access-date=27 December 2013}}&lt;/ref&gt;
2612 The show's set, which for Broadway cost $4.3 million,&lt;ref name=&quot;set&quot;&gt;{{cite web | url=https://nypost.com/2013/10/30/rocky-headed-to-broadway-with-a-4-3m-set/ | title='Rocky' headed to Broadway with a $4.3M set | publisher=[[New York Post]] | work=nypost.com | date=30 October 2013 | access-date=30 December 2013}}&lt;/ref&gt; is set around a mostly bare stage which represents the Gym that Rocky trains in.&lt;ref name = &quot;Singin&quot;/&gt; The production uses sliding box-like sets to represent the homes of the characters&lt;ref name = &quot;Singin&quot;/&gt; and the sliding element goes on to include a full sized regulation [[boxing ring]].&lt;ref name=&quot;theater&quot;&gt;{{cite web | url=http://www.theatermania.com/new-york-city-theater/news/07-2013/for-alex-timbers-directing-theater-is-a-labour-of-_65567.html | title=For Alex Timbers, Directing Theater is a Labo(u)r of Love | publisher=Theatermania | work=theatermania.com | date=17 July 2013 | access-date=30 December 2013}}&lt;/ref&gt; Audience members seated within the front [[Theater (structure)#Basic elements of a theater structure|stalls]] Golden Circle seating section, are escorted onto the stage for the final 20 minutes to sit on [[bleacher]] style seats,&lt;ref name=&quot;info&quot;&gt;{{cite web | url=http://www.rockybroadway.com | title=Tickets and Information |
     publisher=Rocky the Musical ¦ work=rockybroadway.com ¦ access-date=30 December 2013}}&lt;/ref&gt; in doing so this allows the boxing ring to enter the auditorium&lt;ref name = &quot;star&quot;/&gt;
     and sit in rows A-F,&lt;ref name = &quot;info&quot;/&gt; bringing the audience close to the final fight scene.&lt;ref name=&quot;star&quot;&gt;{{cite web ¦
     url=https://www.thestar.com/entertainment/stage/2013/04/29/sylvester_stallones_rocky_turned_into_a_broadway_musical.html ¦ title=Sylvester Stallone's Rocky turned into a Broadway musical ¦
     publisher=The Star ¦ work=thestar.com ¦ date=29 April 2013 ¦ access-date=30 December 2013}}&lt;/ref&gt;
2613 &lt;!-- Quotes along with Yo&lt;ref&gt;{{cite web ¦ url=http://www.broadway.com/buzz/158689/sylvester-stallone-has-big-plans-for-lyrical-and-poetic-rocky-musical/ ¦ title=Sylvester Stallone Has Big
     Plans for 'Lyrical and Poetic' Rocky Musical ¦ publisher=Broadway.Com ¦ work=broadway.com ¦ date=22 November 2011 ¦ access-date=29 December 2013}}&lt;/ref&gt;--&gt;
2614 ''Rocky the Musical'' received its world premiere on 18 November 2012, at the [[Operettenhaus]], [[Hamburg]].&lt;ref name=&quot;premiere&quot;&gt;{{cite web ¦
     url=http://www.torontosun.com/2013/10/17/sylvester-stallone-talks-rocky-musical ¦ title=Sylvester Stallone talks 'Rocky' musical ¦ publisher=[[Toronto Sun]] ¦ work=torontosun.com ¦ date=17 October
     2013 ¦ access-date=22 October 2013}}&lt;/ref&gt; The production opened to positive reviews&lt;ref name=&quot;Singin&quot;&gt;{{cite web ¦ url=https://www.nytimes.com/2012/12/09/theater/a-hit-in-
     germany-a-rocky-musical-aims-at-broadway.html?pagewanted=1 ¦ title=Yo, Adrian! I'm Singin'! ¦ work=[[The New York Times]] ¦ date=5 December 2012 ¦ access-date=27 December 2013}}&lt;/ref&gt; and title
     casting included [[Drew Sarich]] as [[Rocky Balboa]],&lt;ref name=&quot;Welt&quot;&gt;{{cite news ¦ url=https://www.welt.de/regionales/hamburg/article107290169/Hauptdarsteller-von-Rocky-in-Hamburg-
     vorgestellt.html ¦ title=Hauptdarsteller von &quot;Rocky&quot; in Hamburg vorgestellt ¦ publisher=[[Die Welt]] ¦ work=welt.de ¦ date=28 June 2012 ¦ access-date=28 December 2013}}&lt;/ref&gt; Wietske
     van Tongeren playing [[Adrian Pennino]]&lt;ref name = &quot;Welt&quot;/&gt; and Terence Archie playing [[Apollo Creed]].&lt;ref name=&quot;nyt&quot;&gt;{{cite web ¦
     url=http://artsbeat.blogs.nytimes.com/2013/11/06/rocky-on-broadway-finds-its-apollo-creed/ ¦ title='Rocky' on Broadway Finds Its Apollo Creed ¦ work=[[The New York Times]] ¦ date=6 November 2013 ¦
     access-date=29 December 2013}}&lt;/ref&gt; Having been written in [[English language¦English]], the show was translated into [[German language¦German]] for its world premiere.&lt;ref&gt;{{cite web ¦
     url=https://www.theglobeandmail.com/arts/theatre-and-performance/nestruck-on-theatre/rocky-the-musical-heading-to-broadway-by-way-of-germany/article14518730/?page=all ¦ title=Rocky the musical
     heading to Broadway - by way of Germany ¦ publisher=[[The Globe and Mail]] ¦ work=theglobeandmail.com ¦ date=25 September 2013 ¦ access-date=28 December 2013}}&lt;/ref&gt;&lt;ref
     name=&quot;LA&quot;&gt;{{cite web ¦ url=https://articles.latimes.com/2013/nov/06/entertainment/la-et-cm-rocky-broadway-musical-dakin-matthews-stallone-20131105 ¦
     archive-url=https://web.archive.org/web/20131231043345/http://articles.latimes.com/2013/nov/06/entertainment/la-et-cm-rocky-broadway-musical-dakin-matthews-stallone-20131105 ¦ url-status=dead ¦
     archive-date=December 31, 2013 ¦ title=L.A. drama ace Dakin Matthews has singing gig in Broadway 'Rocky' ¦ work=[[Los Angeles Times]] ¦ date=6 November 2013 ¦ access-date=3 January 2014}}&lt;/ref&gt;
2615 ¦ ''[[Reach Me]]'' ¦¦ Vic ¦¦ ¦¦ &lt;ref&gt;{{cite web ¦ url=http://www.cinemablend.com/new/Sylvester-Stallone-Highlights-Huge-Bizarre-Cast-Trailer-Reach-Me-43777.html ¦ title=Sylvester Stallone
     Highlights a Huge Bizarre Cast in Trailer for Reach Me ¦ date=7 July 2014 }}&lt;/ref&gt;
2616 Meyer was born in New York City and raised in Chicago. She graduated from the [[University of Chicago Laboratory Schools¦Chicago Lab School]], [[Grinnell College]], and received an MFA from [[New
     York University]]'s film school. Meyer got her start as an assistant editor on the Oscar-winning documentary ''[[Woodstock (film)¦Woodstock]]'' (1970).&lt;ref name=&quot;Full Cast and Crew
     Credits&quot;&gt;[https://www.imdb.com/title/tt0066580/fullcredits?ref_=tt_ov_st_sm Full Cast &amp; Crew of ''Woodstock''] at the [[Internet Movie
     Database]].&lt;/ref&gt;&lt;ref&gt;[https://www.imdb.com/name/nm0583287/?ref_=ttfc_fc_cr59#editor Editorial Department Credits for Muffie Meyer] at the [[Internet Movie Database]].&lt;/ref&gt; Her
     early editing credits include ''[[The Lords of Flatbush]]'' (1974), starring [[Sylvester Stallone]] and [[Henry Winkler]], and ''[[The Groove Tube]]'' (1974), starring [[Chevy Chase]], a film
     precursor to ''[[Saturday Night Live]]''.
2617 In August 1982, having completing work writing and producing songs for [[Dionne Warwick]]'s 1982 album ''[[Heartbreaker (Dionne Warwick album)¦Heartbreaker]]'', Gibb began work on &quot;[[Eyes That
     See in the Dark (song)¦Eyes That See in the Dark]]&quot;, for [[Kenny Rogers]] after Rogers had earlier requested songs from Gibb. Work on songs from Rogers was interrupted by the production of
     ''[[Staying Alive (1983 film)¦Staying Alive]]'', the 1983 sequel to ''[[Saturday Night Fever]]'' starring [[John Travolta]] and directed by [[Sylvester Stallone]], which was to feature new songs by
     the [[Bee Gees]].&lt;ref&gt;[http://www.columbia.edu/~brennan/beegees/82.html Joseph Brennan - Gibb Songs: 1982]&lt;/ref&gt;
2618 ¦ [[Sylvester Stallone]]
2619 In 1994 Morrell reissued ''The Totem'' in an unexpurgated version.  In his foreword he states that when he submitted the original manuscript to his then-publisher his editor demanded that he simplify
     and shorten the book.  As Morrell had not yet achieved bestseller status (the book was published three years before his 1972 debut novel ''[[First Blood (novel)¦First Blood]]'' was transformed into
     the enormously successful [[Sylvester Stallone]] film), he acquiesced and rewrote the manuscript into a shorter, truncated edition and then simply forgot about the original version.  Morrell claims
     that he came across the original book in the early 1990s and decided to publish it and take the first edition out of circulation.
2620 * Jade Roberts as [[Sylvester Stallone¦Sylvester Stallone Look-a-like]]
2621 Upon arriving in [[Hollywood, Los Angeles¦Hollywood]], Johnson directed ''John G. Avildsen: King of the Underdogs'', about the [[Academy Award]]-winning director whose credits include ''[[Rocky]]''
     and ''[[The Karate Kid (1984 film)¦The Karate Kid]]''.&lt;ref&gt;Kreps, Daniel. [https://www.rollingstone.com/movies/movie-news/john-g-avildsen-rocky-the-karate-kid-director-dead-at-81-199305/
     &quot;John G. Avildsen, 'Rocky,' 'The Karate Kid' Director, Dead at 81&quot;], ''Rolling Stone'', San Francisco, CA, 17 June 2017. Retrieved on 21 August 2018.&lt;/ref&gt; The documentary features
     interviews with [[Sylvester Stallone]], [[Ralph Macchio]], [[Martin Scorsese]], [[Talia Shire]], [[Burt Reynolds]] and many more.&lt;ref&gt;Farber, Stephen.
     [https://www.hollywoodreporter.com/review/john-g-avildsen-king-underdogs-review-973174 &quot;'John G. Avildsen: King of the Underdogs': Film Review¦Santa Barbara 2017&quot;], ''The Hollywood
     Reporter'', Los Angeles, CA, 2 February 2017. Retrieved on 21 August 2018.&lt;/ref&gt; Alongside John Avildsen and the cast and crew of ''[[The Karate Kid (1984 film)¦The Karate Kid]]'', Johnson
     received a Commendation by the [[City of Los Angeles]] for his documentary and its association with ''[[The Karate Kid (1984 film)¦The Karate Kid]]'' legacy. In 2022, Johnson and his documentary were
     mentioned in [[Ralph Macchio]]'s New York Times Bestselling book ''Waxing On: The Karate Kid and Me''.
2622 [[Sylvester Stallone]] hand-picked Johnson to direct ''[[40 Years of Rocky]]'': ''The Birth of a Classic''. The documentary, narrated by Stallone, details the making of ''[[Rocky]]''
     &lt;ref&gt;Drown, Michelle. [http://www.independent.com/news/2017/jan/26/john-avildsen-king-underdogs/ &quot;John G. Avildsen: King of the Underdogs Director Derek Wayne Johnson&quot;], ''The Santa
     Barbara Independent'', Santa Barbara, CA, 26 January 2017. Retrieved on 16 February 2017.&lt;/ref&gt; and was acquired by [[Metro-Goldwyn-Mayer]] (MGM).&lt;ref&gt;Lang,
     Brent.[https://variety.com/2019/film/news/40-years-of-rocky-sylvester-stallone-mgm-1203402764/, &quot;MGM Buying '40 Years of Rocky' to Air on Epix (EXCLUSIVE)&quot;], ''Variety'', Los Angeles, CA,
     13 November 2019. Retrieved on 18 November 2019.&lt;/ref&gt; Johnson and his documentary were mentioned in the 2021 ''[[Life (magazine)¦Life]]'' magazine special issue dedicated to the [[Rocky]]
     franchise.
2623 Johnson directed ''STALLONE: Frank, That Is'', highlighting musician and actor [[Frank Stallone]], brother of Sylvester. The documentary features interviews with [[Sylvester Stallone]], [[Arnold
     Schwarzenegger]], [[Billy Dee Williams]], [[Duff McKagan]], [[Joe Mantegna]] and many others.&lt;ref&gt;Gonzalez, Bobby. [https://ktla.com/2018/02/19/spoken-dreams-derek-wayne-johnson-filmmaker/
     &quot;Spoken Dreams: Derek Wayne Johnson, Filmmaker&quot;], ''KTLA'', Los Angeles, CA, 19 February 2018. Retrieved on 21 August 2018.&lt;/ref&gt;
2624 [[Sylvester Stallone]] once again hand-picked Johnson to edit ''The Making of Rocky vs. Drago'', a documentary directed by [[John Herzfeld]] and distributed by [[Metro-Goldwyn-Mayer]] (MGM).
2625 In the second level of the Allied campaign in [[Yuri's Revenge]] (which takes place in L.A.), there are three special infantry units on the Allied side which are armed with anti-infantry weapons and
     immune to mind control.  Here's my question: When the game first came out, these special units were named &quot;[[Clint Eastwood¦Flint Westwood]]&quot;, &quot;[[Sylvester Stallone¦Sammy
     Stallion]]&quot;, and &quot;[[Arnold Schwarzenegger¦Arnie Frankenfurter]]&quot;, but when the game was reissued as part of &quot;Command and Conquer: The First Decade&quot; collection, they were
     renamed &quot;Cowboy&quot;, &quot;Bodybuilder&quot; and &quot;Hero&quot; respectively.  Why? [[Special:Contributions/24.23.196.85¦24.23.196.85]] ([[User talk:24.23.196.85¦talk]]) 02:43, 17 November
     2013 (UTC)
2626 ::::Well, it's whatever rights are associated with the names &quot;Clint Eastwood&quot;, &quot;Sylvester Stallone&quot; and &quot;Arnold Schwarzenegger&quot;, which will depend on jurisdiction,
     registered trade marks, and many other factors.  The names were changed for _possible_ legal reasons, after all - the games company decided not to take the risk of any of the (potential) rights
     actually being asserted against them. [[User:Tevildo¦Tevildo]] ([[User talk:Tevildo¦talk]]) 21:16, 18 November 2013 (UTC)
2627 :Collect, ''why'' is it likely to be a contentious claim when the sources we're using list '''all''' or '''most''' of the porn films the named actor has appeared in and not just &quot;'''a''' porn
     film&quot;? It's not like we are trying to include [[Sylvester Stalone]] or [[Kristine DeBell]] from [[Meatballs (film)]] in a list of porn actors because they were in a single film. These are people
     who have clearly chosen to be in the Industry. --[[User:Scalhotrod¦Scalhotrod - Just your average banjo playing, drag racing, cowboy...]] ([[User talk:Scalhotrod¦talk]]) 23:54, 29 December 2013 (UTC)
2628 * &quot;Coloured Satellite Launched. Mr. T…I Hope He's Not Reading This!&quot;  Humphries writes on Mr. T (Laurence Tureaud) when he was a bodyguard for Leon Spinks, four years before he played James
     &quot;Clubber&quot; Lang alongside Sylvester Stallone in Rocky III (Rod Humphries Writes from New Orleans, ''The Sun-Herald'', Sydney, 17 September 1978, page 90).
2629 ¦¦ Director: [[Patrick Hughes (director)¦Patrick Hughes]] &lt;br /&gt; Cast: [[Sylvester Stallone]], [[Jason Statham]], [[Antonio Banderas]], [[Jet Li]], [[Wesley Snipes]], [[Dolph Lundgren]],
     [[Kelsey Grammer]], [[Randy Couture]], [[Terry Crews]], [[Kellan Lutz]], [[Ronda Rousey]], [[Glen Powell]], [[Victor Ortiz]], [[Robert Davi]], [[Mel Gibson]], [[Harrison Ford]], [[Arnold
     Schwarzenegger]]
2630 * [[Sylvester Stallone]], actor, filmmaker, screenwriter
2631 Palmisano became known mainly for integrating committee established filmmakers like ''[[First Blood]]'' (1982), ''[[Robocop 2]]'' (1990),  ''[[Batman Forever]]'' (1995), ''[[Free Willy 2]]'' (1995),
     ''[[Assassins (1995 film)¦Assassins]]'' (1995) with [[Sylvester Stallone]], ''[[Rush Hour 2]]'' (2001) with [[Jackie Chan]]; ''[[After the Sunset]]'' (2004) with [[Pierce Brosnan]] and [[Salma
     Hayek]], ''[[The Other Guys]]'' (2010) with [[Will Ferrell]], among others. He was nominated for a [[Taurus World Stunt Awards¦Taurus Award]] for both ''[[Rush Hour 2]]'' (2001) and ''[[Rush Hour
     3]]'' (2007).&lt;ref&gt;{{Cite web¦title = IMDB - Conrad E. Palmisano - Awards &amp; Nominations¦ website=[[IMDb]] ¦url = https://www.imdb.com/name/nm0003886/awards?ref_=nm_awd ¦accessdate = 2015-07-
     12}}&lt;/ref&gt;
2632 * 1996 ''[[Daylight (1996 film)¦Daylight]]'': Dir. [[Rob Cohen]], Universal Pictures with [[Sylvester Stallone]] and [[Viggo Mortensen]].
2633 ** [[Sylvester Stallone]] in ''[[Bullet to the Head]]'', ''[[Escape Plan (film)¦Escape Plan]]'' and ''[[Grudge Match]]'' as James &quot;Bobo&quot; Bonomo, Ray Breslin, and Henry &quot;Razor&quot;
     Sharp (respectively)
2634 Hughes' various commercials and previous 2008 [[short film]] ''[[Signs (short film)¦Signs]]''&lt;ref&gt;{{Cite news¦url=http://www.smh.com.au/entertainment/movies/disney-sweeps-in-to-collect-the-
     oscars-glory-but-film-fans-reckon-theyve-seen-it-all-before-20130227-2f6e3.html¦title=Disney sweeps in to collect the Oscars glory but film fans reckon they've seen it all before¦last=Quinn¦
     first=Karl¦newspaper=The Sydney Morning Herald¦language=en-US¦access-date=2016-10-07}}&lt;/ref&gt; caught the attention of Hollywood star [[Sylvester Stallone]], who was looking for 'new blood' to
     direct the third installment of the lucrative ''[[The Expendables (franchise)¦The Expendables]]'' franchise.&lt;ref name=SS&gt;{{cite web¦title=The Sly Stallone Official
     Twitter¦url=https://twitter.com/TheSlyStallone¦publisher=[[Twitter]]¦accessdate=20 April 2013}} {{fcn¦date=November 2019}}&lt;/ref&gt; Hughes filmed ''[[The Expendables 3]]'' from August to October
     2013 and the film held its world premiere in London on 4 August 2014.
2635 [[IGN]] gave the film a score of 6.0/10, saying, &quot;''Hitman: Agent 47'' is almost certainly going to be too much of a generic action film for those heavily invested in the game franchise, and too
     video game-like for those who aren't.&quot;&lt;ref&gt;{{cite web¦url=https://www.ign.com/articles/2015/08/19/hitman-agent-47-review¦title=Hitman: Agent 47 Review¦author=Josh Lasser¦date=August 19,
```



```
2015|work=IGN|access-date=April 17, 2020|archive-date=May 23, 2019|archive-url=https://web.archive.org/web/20190523015324/https://www.ign.com/articles/2015/08/19/hitman-agent-47-review|url-status=live}}&lt;/ref&gt; IrishFilmCritic gave the film 3.5/5 stars, describing the target audience as &quot;those of us who grew up in the 70’s and 80’s and thrived on overly exaggerated action films with anything that starred [[Arnold Schwarzenegger]], [[Sylvester Stallone]] and [[Bruce Willis]]... Go to this movie and just have fun, it’s that simple.&quot;&lt;ref&gt;{{cite web|url=http://irishfilmcritic.com/movie-review-hitman-agent-47-is-ridiculously-excessive-fun/|title=Movie Review: &quot;Hitman: Agent 47&quot; Is Ridiculously Excessive Fun|author=James McDonald|date=August 20, 2015|work=Irish Film Critic|access-date=August 21, 2015|archive-date=January 7, 2016|archive-url=https://web.archive.org/web/20160107175042/http://irishfilmcritic.com/movie-review-hitman-agent-47-is-ridiculously-excessive-fun/|url-status=live}}&lt;/ref&gt; [[Kotaku]] also gave the film a positive review.&lt;ref&gt;{{Cite web|url = http://kotaku.com/i-saw-hitman-agent-47-and-it-was-fine-1725968305|title = I Saw Hitman: Agent 47, and It Was Fine|date = August 23, 2015|website = Kotaku|last = Owen|first = Phil|access-date = August 24, 2015|archive-date = August 26, 2015|archive-url = https://web.archive.org/web/20150826002017/http://kotaku.com/i-saw-hitman-agent-47-and-it-was-fine-1725968305|url-status = live}}&lt;/ref&gt;
2636 |Guests=[[Sylvester Stallone]], [[Missy Franklin]]
2637 |Guests=[[Sylvester Stallone]], [[Judd Apatow]]
2638 * [[Sylvester Stallone]]
2639 At the age of twelve, Bleck’s mother bought her and her sister their first weight bench. She cites actor [[Sylvester Stallone]]’s character [[Rocky Balboa]] and the [[Hulk|Incredible Hulk]] as her early inspirations. Bleck, by that time a modern dancer, would not begin lifting seriously until the age of 16, when she joined a powerlifting. Her first trainer was Jake Grabow of Better Builds Gym, from whom she received an education on eating and posing. Not long after high school graduation, Bleck started competing as a bodybuilder.
2640 Carey regretted not visiting China before the script was written, as it caused problems during the filming of the village scenes in the [[Hubei|Hubei province]].&lt;ref name=&quot;Leicester&quot;/&gt;&lt;ref name=&quot;Streets&quot;/&gt; As they had imagined what a Chinese village would look like, it meant that they had to [[Set redress|redress]] the actual village in which they filmed.&lt;ref name=&quot;Leicester&quot;/&gt; After discovering [[Tiananmen Square]] had been closed off for renovations, the crew were forced to film scenes in a side street.&lt;ref name=&quot;Leicester&quot;/&gt; Other locations used during filming included the [[Great Wall of China]] and the [[Forbidden City]].&lt;ref name=&quot;ITN&quot;/&gt; Chinese extras were also employed during the shoot.&lt;ref name=&quot;Streets&quot;/&gt; While filming the final scene of the episode, Carey laid down on a stone carving at the [[Temple of Heaven]], causing the Chinese police to halt the shoot and issue him with a fine.&lt;ref name=&quot;Streets&quot;/&gt; Of the situation, Cohen commented &quot;I guess the system is different. We thought, ‘OK, we had a permit for this, we are ready for this, we are going to be here on a certain day at a certain time and everybody knows, right?’&quot;&lt;ref name=&quot;Streets&quot;/&gt; The majority of the scene had already been filmed and Carey was not made to pay the fine.&lt;ref name=&quot;Streets&quot;/&gt; The finale is an imitation of [[Rocky Balboa]]’s ([[Sylvester Stallone]]) run up the steps of the [[Philadelphia Museum of Art]] in ’’[[Rocky]]’’.&lt;ref name=&quot;Leicester&quot;/&gt;
2641 |[[Sylvester Stallone]]
2642 |First collaboration with Sylvester Stallone
2643 |Sylvester Stallone
2644 |Second collaboration with Sylvester Stallone
2645 He was also cinematographer on well-known films such as [[Sylvester Stallone]]’s ’’[[Lock Up (1989 film)|Lock Up]]’’ and ’’[[Tango &amp; Cash]]’’, [[Whoopi Goldberg]]’s ’’[[Boys on the Side]]’’, [[Bette Midler]]’s ’’[[The First Wives Club]]’’, and [[Samuel L. Jackson]]’s ’’[[Shaft (2000 film)|Shaft]]’’.
2646 Jelly Roll made his television debut on the television drama series ’’[[Tulsa King]]’’ with [[Sylvester Stallone]]. The episode was released in September 2024 and featured Jelly Roll performing a rendition of &quot;[[I Am Not Okay]]&quot;.&lt;ref&gt;{{cite web|url=https://www.usatoday.com/story/entertainment/tv/2024/09/20/jelly-roll-tulsa-king-appearance/75295227007/|title=Jelly Roll makes ’Tulsa King’ TV debut with Sylvester Stallone’s mobster: Watch them meet|last=Alexander|first=Bryan|date=September 20, 2024|accessdate=September 22, 2024|work=[[USA Today]]}}&lt;/ref&gt;
2647 |Lifetime Achievement Award || colspan=2|[[Sylvester Stallone]] ||
2648 |[[Ryan Coogler]] (director/screenplay); [[Aaron Covington]] (screenplay); [[Michael B. Jordan]], [[Sylvester Stallone]], [[Tessa Thompson]], [[Phylicia Rashad]], [[Tony Bellew|Anthony Bellew]]
2649 In 2019 [[Universal Pictures]] released ’I Am Durán’ - Hodgson’s documentary film about Panamanian boxer [[Roberto Durán]]. The focus of the film was to chart Durán’s rise in the sport set against the backdrop of social and political unrest in his homeland,&lt;ref&gt;{{cite web|last1=Jones|first1=Rich|date=8 June 2019|title=I Am Duran documentary provides unique insight into Panama icon|url=https://www.mirror.co.uk/sport/boxing/boxing-news-roberto-duran-documentary-16482832|website=The Mirror|language=en}}&lt;/ref&gt; [[Panama]]. The film features [[Roberto Durán]], [[Robert De Niro]], [[Sylvester Stallone]], [[Mike Tyson]], General [[Manuel Noriega]], [[Sugar Ray Leonard]], [[Marvelous Marvin Hagler]], [[Don King (boxing promoter)|Don King]], [[Bob Arum]], [[Ricky Hatton]], [[Lennox Lewis]] and [[Oscar De La Hoya]] amongst others.&lt;ref&gt;{{cite web|last1=Mazique|first1=Brian|title=’I Am Duran’ Review: Roberto Duran Documentary Is A Film Most Boxing Fans Will Enjoy|url=https://www.forbes.com/sites/brianmazique/2019/06/08/i-am-duran-review-roberto-duran-documentary-is-a-film-most-boxing-fans-will-enjoy/?sh=2c64878b3c9e|website=Forbes|language=en}}&lt;/ref&gt;  Noriega’s interview was his last ever broadcast interview before he passed away. The film reviewed favourably and once again Hodgson was shortlisted for a [[Grierson Awards|Grierson Award]] - this time in ‘Best International Documentary’ category. It saw off competition from the highly acclaimed film ‘[[Diego Maradona (film)|Diego Maradona]]’ by Oscar winner [[Asif Kapadia]] to scoop a [[FOCAL International|Focal International Award]] in 2020, in the ‘Sports Production’ category.
2650 In 2016, it was announced that Mockridge would be a host on the [[Netflix]] reality show ’’[[Ultimate Beastmaster]]’’.&lt;ref&gt;{{cite web|url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|title=Netflix Orders Global Competition Series ’Ultimate Beastmaster’ from Sylvester Stallone, Dave Broome|last=Prudon|first=Laura|work=Variety|date=9 May 2016|accessdate=5 October 2016 |language=en}}&lt;/ref&gt;
2651 Also the apparel line has gained the attention of rock stars and musicians, including: [[Steve Vai]], [[Dylan McDermott]], [[Sylvester Stallone]], [[Orlando Bloom]], [[Steve Lukather]] of [[Toto (band)|Toto]], [[Robert Trujillo]] of [[Metallica]], [[Usher (entertainer)|Usher]], [[Chris Brown]], [[Future (rapper)|Future]], Nicole Murphy, [[Columbus Short]], and [[RJ Mitte]].&lt;ref name=&quot;Dark Beauty&quot; /&gt;
2652 * ’’[[D-Tox]]’’, a 2002 American psychological thriller horror film directed by Jim Gillespie and starring Sylvester Stallone.
2653 The Hoyt Buffalo hunting recurve was used by the character Hawkeye in the [[The Avengers (2012 film)|’’Avengers’’]] movie, as well as by Katniss Everdeen in ’’[[The Hunger Games (film series)|The Hunger Games]]’’ series.&lt;ref&gt;{{Cite web |url=http://www.hoyt.com/community/news_detail.php?id=505 |title=Jennifer Lawrence Shoots Hoyt in Hunger Games |access-date=2014-05-13 |archive-url=https://web.archive.org/web/20140325032441/http://www.hoyt.com/community/news_detail.php?id=505 |archive-date=2014-03-25 |url-status=dead }}&lt;/ref&gt; The Hoyt Gamemaster II was used by Hawkeye in ’’[[Avengers: Age of Ultron]]’’. The Hoyt/Spectra bow was the principal silent weapon used by [[Sylvester Stallone]] as John Rambo in ’’[[Rambo: First Blood Part II]]’’ and ’’[[Rambo III]]’’. In the backstory Rambo had attained skill with a bow since childhood. In the film he uses arrows tipped with explosive warheads.
2654 ’’United Passions’’ received the Barry L. Bumstead Award during the [[36th Golden Raspberry Awards]],&lt;ref&gt;{{Cite web|url=https://www.hollywoodreporter.com/news/razzie-awards-mock-fifty-shades-870615|title=Razzie Awards Mock ’Fifty Shades of Grey,’ Redeem Sylvester Stallone|website=The Hollywood Reporter|date=28 February 2016|language=en|access-date=2020-01-06}}&lt;/ref&gt; a special category for critical and financial failures that were not given an eligible release.&lt;ref&gt;{{Cite web|url=https://www.smh.com.au/entertainment/movies/razzie-awards-hillarys-america-blows-golden-raspberries-at-preoscar-awards-shamefest-20170226-gulepe.html|title=Razzie Awards: Hillary’s America blows golden raspberries at pre-Oscar awards shamefest|last=Idato|first=Michael|date=2017-02-26|website=The Sydney Morning Herald|access-date=2020-01-06|archive-date=21 June 2020|archive-url=https://web.archive.org/web/20200621183541/https://www.smh.com.au/entertainment/movies/razzie-awards-hillarys-america-blows-golden-raspberries-at-preoscar-awards-shamefest-20170226-gulepe.html|url-status=live}}&lt;/ref&gt;
2655 She worked on several films with director [[Sidney Lumet]], including ’’[[Network (1976 film)|Network]]’’,&lt;ref name=thr&gt;{{cite web|url=http://www.hollywoodreporter.com/news/casting-director-joy-todd-dies-682584|title=Casting Director Joy Todd Dies|publisher=The Hollywood Reporter|accessdate=2014-07-16}}&lt;/ref&gt; ’’[[Prince of the City]]’’, ’’[[The Verdict (1982 film)|The Verdict]]’’, ’’[[Garbo Talks]]’’,&lt;ref name=var&gt;{{cite web|url=https://variety.com/2014/film/people-news/joy-todd-casting-director-for-sidney-lumet-dies-1201116713|title=Joy Todd, Casting Director for Sidney Lumet, Dies|publisher=Variety|accessdate=2014-07-16}}&lt;/ref&gt; ’’[[Family Business (1989 film)|Family Business]]’’, and ’’[[Q&amp;A (film)|Q&amp;A]]’’. She also worked on several films starring [[Sylvester Stallone]], including ’’[[Rocky II]]’’, ’’[[Rocky III]]’’, ’’[[Cobra (1986 film)|Cobra]]’’, ’’[[Rambo III]]’’, ’’[[Lock Up (1989 film)|Lock Up]]’’, and ’’[[Demolition Man (film)|Demolition Man]]’’.
2656       <text bytes="20479" sha1="5l5uyjspqmq34zieo0k5uldb51ha757" xml:space="preserve">’’’Jeff Locker’’’ (’’’傑夫’’’&lt;ref&gt;{{cite web|last1=Ye|first1=JunYuan|title=外國人過年》傑夫犯太歲動念回美國|url=http://news.sina.com/udn/201-104-103-109/2006-01-27/1221595731.html|website=news.sina.com/|publisher=SINA Corporation|access-date=17 June 2014|archive-date=2016-03-03|archive-url=https://web.archive.org/web/20160303234642/http://news.sina.com/udn/201-104-103-109/2006-01-27/1221595731.html|url-status=dead}}&lt;/ref&gt;) is an American actor, playwright, screenwriter, host, and author. He appeared on Marvel’s ’’[[Agent Carter (TV series)|Agent Carter]]’’,&lt;ref&gt;{{cite web | url=http://marvel.com/news/tv/23802/marvels_agent_carter_debriefs_her_first_2_missions | title=Marvel News, Blog, Articles &amp; Press Releases &amp;#124; Marvel |}}&lt;/ref&gt; in feature films ’’Stasis’’, Warner Bros. release ’’Disaster L.A.’’&lt;ref&gt;{{cite web | url=http://www.shocktillyoudrop.com/news/358077-warner-bros-home-entertainment-picks-up-disaster-la/ | title=Horror Archives | date=11 August 2023 }}&lt;/ref&gt; and Taiwanese box office smash ’’[[Formula 17]]’’ (17 歲的天空), in multiple sketches on ’’[[Jimmy Kimmel Live!]]’’, and starred as Peter Timms in the play ’’[[Chinglish (play)|Chinglish]]’’&lt;ref&gt;{{cite journal|last1=Tucker|first1=Nathan|title=Review: PCS’s &quot;Chinglish&quot;|journal=Portland Monthly Mag|date=January 23, 2014|url=http://www.portlandmonthlymag.com/arts-and-entertainment/culturephile-portland-arts/articles/review-pcss-chinglish-january-2014|access-date=June 17, 2014|archive-date=July 14, 2014|archive-url=https://web.archive.org/web/20140714172709/http://www.portlandmonthlymag.com/arts-and-entertainment/culturephile-portland-arts/articles/review-pcss-chinglish-january-2014|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite web| url = http://www.broadwayworld.com/portland/article/BWW-Reviews-CHINGLISH-is-Filled-with-Cross-Cultural-Laughter-at-Portland-Center-Stage-20140128#.U5_PjI1dWyM| title = Reviews: CHINGLISH is Filled with Cross-Cultural Laughter at Portland Center Stage}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |url=http://www.wwnytv.com/features/craigs-to-do-list/Chinglish-at-Syracuse-Stage--249016081.html |title=Chinglish at Syracuse Stage &amp;#124; WWNY TV 7 - News, Weather and Sports for &amp;#124; Craig’s to do List |access-date=2014-06-17 |archive-url=https://web.archive.org/web/20140714201034/http://www.wwnytv.com/features/craigs-to-do-list/Chinglish-at-Syracuse-Stage--249016081.html |archive-date=2014-07-14 |url-status=dead }}&lt;/ref&gt; by [[David Henry Hwang]]. Among his several award-winning plays, The Forgotten Place won the 43rd [[Samuel French, Inc.|Samuel French]] [[Off-Off-Broadway Short Play Festival|Off Off Broadway Short Play Festival]]&lt;ref&gt;{{Cite web|url=http://www.playbill.com/article/helen-park-gracie-gardner-and-more-named-winners-of-samuel-frenchs-off-off-broadway-short-play-festival|title=Helen Park, Gracie Gardner, and More Named Winners of Samuel French’s Off Off Broadway Short Play Festival|last=Clement|first=Olivia|date=2018-08-27|website=Playbill|language=en|access-date=2019-12-22}}&lt;/ref&gt; and is published and licensed worldwide by Samuel French. The short film adaption of The Forgotten Place won 21 awards during its film festival run.&lt;ref&gt;{{Cite web |last=Wild |first=Stephi |title=VOODOO MACBETH, THE FORGOTTEN PLACE, ON OUR OWN ISLAND, and More Win Big at the 27th Annual Sedona International Film Festival |url=https://www.broadwayworld.com/bwwtv/article/VOODOO-MACBETH-THE-FORGOTTEN-PLACE-ON-OUR-OWN-ISLAND-and-More-Win-Big-at-the-27th-Annual-Sedona-International-Film-Festival-20210620 |access-date=2022-08-15 |website=BroadwayWorld.com |language=en}}&lt;/ref&gt; As a screenwriter, he made [[The Black List (survey)|The Black List]]’s [[GLAAD]] List &lt;ref&gt;{{Cite web |last1=Donnelly |
```

first1=Matt |date=2022-04-20 |title=The GLAAD List: Best Unproduced LGBTQ Screenplays of 2022 Unveiled (EXCLUSIVE) |url=https://variety.com/2022/film/news/glaad-black-list-2022-screenplays-margot-robbie-1235236322/ |access-date=2022-08-15 |website=Variety |language=en-US}}&lt;/ref&gt; and won the [[Atlanta Film Festival]] Screenplay Competition.&lt;ref&gt;{{Cite web |last=Pryce |first=Kevon |title=2022 Atlanta Film Festival Screenplay Competition Winners |url=https://www.atlantafilmfestival.com/atlff-news/2022/4/14/2022-atlanta-film-festival-screenplay-competition-winners |access-date=2022-08-15 |website=Atlanta Film Festival |date=14 April 2022 |language=en-US}}&lt;/ref&gt; Fluent in Mandarin Chinese, he was the host of the 75th Golden Globe Awards Red Carpet Show,&lt;ref&gt;{{Citation|title=STX Tencent 75th Golden Globe Awards Red Carpet Show|url=http://www.imdb.com/title/tt7929010/|access-date=2019-12-22}}&lt;/ref&gt; game show ''Who's Smart'' 金頭腦 on [[ETTV America]],&lt;ref&gt;{{cite web | url=http://dailynews.sina.com/bg/news/usa/sinaus/su/20150323/17076543351.html | title=新闻中心首页_新浪网}}&lt;/ref&gt; Sony AXN's Fear Challenge (台灣誰敢來挑戰, a Chinese version of ''[[Fear Factor]]''),&lt;ref&gt;{{cite web|last1=Phips|first1=Gavin|title=Face the terror at Taiwan's Fear Challenge 2003|url=http://www.taipeitimes.com/News/feat/archives/2003/11/21/2003076738|website=taipeitimes.com|date=21 November 2003 |publisher=Liberty Times|access-date=18 June 2014}}&lt;/ref&gt; as well as a host at the [[Huading Awards]],&lt;ref&gt;{{Cite web|url=https://www.hollywoodreporter.com/news/why-a-little-known-chinese-awards-show-draws-likes-natalie-portman-mel-gibson-sylvester-stallon|title=Why a Little-Known Chinese Awards Show Draws the Likes of Natalie Portman, Mel Gibson and Sylvester Stallone|website=The Hollywood Reporter|date=18 December 2016 |language=en|access-date=2019-12-22}}&lt;/ref&gt; [[Golden Horse Awards]] and [[Golden Bell Awards]].&lt;ref&gt;{{cite web|last1=Wang|first1=FeiHua|title=八十九年廣播金鐘獎頒獎典禮主持人名單十三日公布|url=http://nrch.cca.gov.tw/ccahome/photo/photo_meta.jsp?pictureurl=cca220002-hp-2000031303490000001-0001-t.jpg&amp;xml_id=0000507097&amp;collectionname=%E4%B8%AD%E5%A4%AE%E7%A4%BE%E8%80%81%E7%85%A7%E7%89%87%E8%A8%88%E5%8A%83&amp;topicname=|website=nrch.cca.gov.tw/|publisher=Taiwan Ministry of Culture|access-date=18 June 2014|archive-url=https://archive.today/20140618062736/http://nrch.cca.gov.tw/ccahome/photo/photo_meta.jsp?pictureurl=cca220002-hp-2000031303490000001-0001-t.jpg&amp;xml_id=0000507097&amp;collectionname=%E4%B8%AD%E5%A4%AE%E7%A4%BE%E8%80%81%E7%85%A7%E7%89%87%E8%A8%88%E5%8A%83&amp;topicname=|archive-date=18 June 2014|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=2000 年廣播金鐘獎 阿扁總統頒發「特別獎」給崔小萍|url=http://www.hi-on.org.tw/bulletins.jsp?b_ID=48029|website=hi-on.org.tw/|access-date=18 June 2014|archive-date=24 September 2015|archive-url=https://web.archive.org/web/20150924030118/http://www.hi-on.org.tw/bulletins.jsp?b_ID=48029|url-status=dead}}&lt;/ref&gt; He also published eight bestselling books on learning English and understanding American culture&lt;ref&gt;{{cite web|title=傑夫英語猛男秀|url=http://www.books.com.tw/products/0010122796|website=books.com.tw/|publisher=Books.Com.Tw|access-date=17 June 2014}}&lt;/ref&gt; and has lectured extensively throughout Taiwan and China.&lt;ref&gt;{{cite web|last1=Zhao|first1=PuSen|title=DJ 傑夫教英文 正修科大「笑」果十足|url=http://www.english.com.tw/modules/news/article.php?storyid=379|website=english.com.tw/|publisher=English.Com.Tw|access-date=17 June 2014|archive-date=10 November 2022|archive-url=https://web.archive.org/web/20221110103811/https://www.english.com.tw/modules/news/article.php?storyid=379|url-status=dead}}&lt;/ref&gt;

2657 On 10 July 200 people voiced support for Israel at the [[Rockland County Courthouse and Dutch Gardens|Rockland County Courthouse]].&lt;ref&gt;{{cite web|last=O'Rourke |first=James|url=http://www.lohud.com/story/news/local/rockland/2014/07/10/pro-israel-rally-held-rockland-hostilities-mount/12507207/|title=Pro-Israel rally held in Rockland as hostilities mount|access-date=12 July 2014|newspaper=The Journal News}}&lt;/ref&gt; On 11 July, In [[Philadelphia]], Israel supporters rallied outside the Israeli consulate.&lt;ref&gt;{{cite web|last=Tawa |first=Steve|url=http://philadelphia.cbslocal.com/2014/07/11/israel-supporters-rally-in-philadelphia-as-gaza-conflict-boils-over/|title=Israel Supporters Rally in Philadelphia As Gaza Conflict Boils Over|access-date=12 July 2014|publisher=CBS Philly|date=11 July 2014}}&lt;/ref&gt; A plethora of actors, producers, directors and other entertainment professionals signed a statement opposing Hamas.&lt;ref&gt;{{cite news|title=Close To 200 Hollywood Execs, Stars Sign Statement Against Hamas: Sylvester Stallone, Amy Pascal, Seth Rogen, Richard Plepler, Sherry Lansing|url=https://deadline.com/2014/08/sylvester-stallone-seth-rogen-avi-lerner-sherry-lansing-among-190-names-sigining-petition-against-hamas-823777/|work=Deadline.com}}&lt;/ref&gt; A pro-Israel rally of more than 1,200 supporters in Los Angeles resulted in physical violence after its protesters ripped a Palestinian flag away from a passing vehicle and began stomping on it.&lt;ref name=&quot;LA&quot;&gt;{{cite news|title=Fight at pro-Israel rally started over stomping of Palestinian flag|url=https://latimes.com/local/lanow/la-me-ln-israel-palestine-federal-building-20140714-story.html|work=Los Angeles Times}}&lt;/ref&gt; On 13 July, a pro-Israel rally attacked a passing car that was flying Palestinian flags in [[Westwood, Los Angeles]]. A police officer fired a shot to intervene in the situation.&lt;ref&gt;{{cite web|url=https://latimes.com/local/lanow/la-me-ln-israel-palestine-federal-building-20140714-story.html|title=Fight at pro-Israel rally started over stomping of Palestinian flag|author=Los Angeles Times|date=14 July 2014|work=Los Angeles Times}}&lt;/ref&gt; On 14 July, over two dozen elected officials rallied at [[New York City Hall]] in support of Israel.&lt;ref name=&quot;NYC&quot;&gt;{{cite news|title=NY Elected Officials Hold Rally For Israel As Pro-Palestinian Protesters Stage Counter-Demonstration|url=http://newyork.cbslocal.com/2014/07/14/ny-elected-officials-hold-rally-for-israel-as-pro-palestinian-protesters-stage-counter-demonstration/|date=14 July 2014}}&lt;/ref&gt; On 17 July over 1,200 supporters attended a demonstration at a local synagogue in the [[Greater Toronto area]].&lt;ref name=&quot;CJNEWS&quot;&gt;{{Cite web|url=http://www.cjnews.com/node/127543|title=Over 1,000 gather at Beth Tzedec for pro-Israel rally - The Canadian Jewish News}}&lt;/ref&gt; Around 5,000 people attended a 20 July pro-Israel rally in New York City, while a smaller counter-protest was held nearby. Both rallies were held without incident.&lt;ref&gt;{{cite web|url=http://westchester.news12.com/news/israel-supporters-rally-in-times-square-1.8851644|title=Israel supporters rally in Times Square|date=20 July 2014|publisher=News 12 Westchester|access-date=21 July 2014|url-status=dead|archive-url=https://web.archive.org/web/20140724072251/http://westchester.news12.com/news/israel-supporters-rally-in-times-square-1.8851644|archive-date=24 July 2014}}&lt;/ref&gt;

2658 Does anyone know of a web-site (or some other source) that collectively lists salaries for various actors and celebrities in various films, TV shows, theatre roles, etc.? I know that this information can be found piece-meal here and there. I am wondering if there is some sort of collective list of information, all found in one place. For example, it would state: Alan Alda was paid $3 million per episode on the TV show ''M*A*S*H''; Sylvester Stallone was paid $28 million for his role in the film ''Rocky''; Nathan Lane was paid $11 million for his role on Broadway in ''The Producers''.   Stuff like that.   Thanks.   [[User:Joseph A. Spadaro|Joseph A. Spadaro]] ([[User talk:Joseph A. Spadaro|talk]]) 18:21, 14 June 2014 (UTC)

2659 [[Mike Myers]] directed ''[[Supermensch: The Legend of Shep Gordon]]'', a documentary on Shep's life, distributed theatrically by [[The Weinstein Company]] in 2013.&lt;ref name=spm&gt;Amy Kaufman, [http://www.latimes.com/entertainment/movies/moviesnow/la-et-mn-mike-myers-shep-gordon-20130722,0,3916923.story#axzz3076PSVDd &quot;Mike Myers to direct documentary about talent manager Shep Gordon&quot;], ''[[Los Angeles Times]]'', July 22, 2013.&lt;/ref&gt; It featured [[Michael Douglas]], [[Sylvester Stallone]], [[Alice Cooper]], [[Steven Tyler]], [[Willie Nelson]] and [[Sammy Hagar]].&lt;ref name=spm /&gt;

2660 |Guests=[[Sylvester Stallone]], [[Eve Hewson]]

2661 In 1987, she partnered up with Jeff Kleiser to form Kleiser-Walczak Construction Company.&lt;ref&gt;{{cite web|title=Bringing Spidey to Life: Kleiser-Walczak Construction Company|url=http://www.awn.com/animationworld/bringing-spidey-life-kleiser-walczak-construction-company|publisher=awn.com|accessdate=14 August 2014}}&lt;/ref&gt; This company is now called Synthespian Studios.&lt;ref&gt;{{cite web|title=Academy Explores &quot;The Development of the Digital Animator&quot; On May 21|date=2 May 2012 |url=http://www.wearemoviegeeks.com/2012/05/academy-explores-the-development-of-the-digital-animator-on-may-21/|publisher=wearemoviegeeks.com|accessdate=14 August 2014}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=Radio City unwraps new Christmas film|date=22 November 2007 |url=http://www.hollywoodreporter.com/news/radio-city-unwraps-new-christmas-155620|publisher=hollywoodreporter.com|accessdate=14 August 2014}}&lt;/ref&gt; They developed some of the first computer-generated humans. They created the first digital [[stunt doubles]] for the feature film, [[Judge Dredd (film)|Judge Dredd]] starring [[Sylvester Stallone]]. They also created the first face replacements in a feature film in ''[[The One (2001 film)|The One]]'' starring [[Jet Li]]. Some additional feature films in which her work can be seen include ''[[Stargate (film)|Stargate]]'', ''[[X-Men (film)|X-Men]]'', ''[[Fantastic Four (2005 film)|Fantastic Four]]'', ''[[Surrogates (film)|Surrogates]]'' and she worked on the animated [[Columbia Pictures]] film logo.&lt;ref&gt;{{cite web|title=DIANA WALCZAK|url=http://www.evo1.org/dianawalczak.html|publisher=evo1.org|accessdate=14 August 2014}}&lt;/ref&gt;

2662 * [[Sylvester Stallone]]

2663 '''''Guardians of the Galaxy Vol. 2''''' is a 2017 American [[superhero film]] based on the [[Marvel Comics]] superhero team [[Guardians of the Galaxy (2008 team)|Guardians of the Galaxy]], produced by [[Marvel Studios]] and distributed by [[Walt Disney Studios Motion Pictures]]. It is the sequel to ''[[Guardians of the Galaxy (film)|Guardians of the Galaxy]]'' (2014) and [[List of Marvel Cinematic Universe films|the 15th film]] in the [[Marvel Cinematic Universe]] (MCU). Written and directed by [[James Gunn]], the film stars an [[ensemble cast]] featuring [[Chris Pratt]], [[Zoe Saldaña]], [[Dave Bautista]], [[Vin Diesel]], [[Bradley Cooper]], [[Michael Rooker]], [[Karen Gillan]], [[Pom Klementieff]], [[Sylvester Stallone]], and [[Kurt Russell]]. In the film, the [[Guardians of the Galaxy (Marvel Cinematic Universe)|Guardians]] travel throughout the cosmos as they help [[Peter Quill (Marvel Cinematic Universe)|Peter Quill]] (Pratt) learn more about his mysterious parentage.

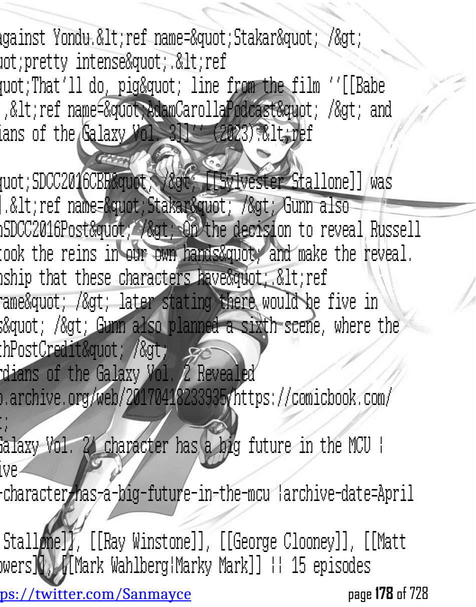
2664 * [[Sylvester Stallone]] as [[Stakar Ogord (Marvel Cinematic Universe)|Stakar Ogord]]:&lt;br /&gt;A high-ranking Ravager who holds a grudge against Yondu.&lt;ref name=&quot;Stakar&quot; /&gt; Stallone likened his character's relationship with Yondu to a father-son relationship, and called the confrontation they have in the film &quot;pretty intense&quot;.&lt;ref name=&quot;PressNotes&quot; /&gt; For Stakar's acceptance of Yondu as a Ravager at the end of the film, Gunn asked Stallone to channel the &quot;That'll do, pig&quot; line from the film ''[[Babe (film)|Babe]]''.&lt;ref name=&quot;58ThingsCommentary2&quot; /&gt; Gunn described Stakar as &quot;very important to the Marvel Universe&quot;,&lt;ref name=&quot;AdamCarollaPodcast&quot; /&gt; and said that &quot;it's our plan to see more of Stallone&quot; in future MCU films, though he was not sure then if that would include ''[[Guardians of the Galaxy Vol. 3]]'' (2023).&lt;ref name=&quot;OldTeam&quot; /&gt;

2665 At San Diego Comic-Con in 2016, Russell and Debicki were revealed to be playing Ego, Quill's father, and Ayesha, respectively.&lt;ref name=&quot;SDCC2016CBR&quot; /&gt; [[Sylvester Stallone]] was also revealed to be in the film,&lt;ref name=&quot;SDCC2016CBR&quot; /&gt; with his role later revealed as [[Starhawk (comics)|Stakar Ogord]].&lt;ref name=&quot;Stakar&quot; /&gt; Gunn also introduced multiple actors who were playing Ravager characters, since the Ravagers have a larger presence in the film.&lt;ref name=&quot;GunnSDCC2016Post&quot; /&gt; On the decision to reveal Russell as Ego and Quill's father when he did, Gunn felt that since &quot;people were going to figure it out eventually [...] it was better that we took the reins in our own hands&quot; and make the reveal. Gunn also felt that this would move the focus of audiences from wanting to discover who Quill's father is to &quot;the story and the relationship that these characters have&quot;.&lt;ref name=&quot;GunnEgoReveal&quot; /&gt; In August 2016, Gunn confirmed the film would feature a [[post-credits scene]],&lt;ref name=&quot;TimeFrame&quot; /&gt; later stating there would be five in total, with four mid-credits scenes and one post-credits.&lt;ref name=&quot;5PostCreditScenes&quot; /&gt;&lt;ref name=&quot;PostCreditsScenes&quot; /&gt; Gunn also planned a sixth scene, where the Ravager Gef is found &quot;mortally wounded&quot; on the ship, but &quot;it ended up being a little confusing&quot;.&lt;ref name=&quot;Gunn6thPostCredit&quot; /&gt;

2666 &lt;ref name=&quot;Stakar&quot;&gt;{{Cite web |last=Peters |first=Megan |date=April 17, 2017 |title=Sylvester Stallone's Mystery Role in Guardians of the Galaxy Vol. 2 Revealed |url=https://comicbook.com/marvel/2017/04/17/guardians-of-the-galaxy-vol-2-sylvester-stallone-role/ |url-status=live |archive-url=https://web.archive.org/web/20170418233935/https://comicbook.com/marvel/2017/04/17/guardians-of-the-galaxy-vol-2-sylvester-stallone-role/ |archive-date=April 18, 2017 |website=[[ComicBook.com]]}}&lt;/ref&gt;

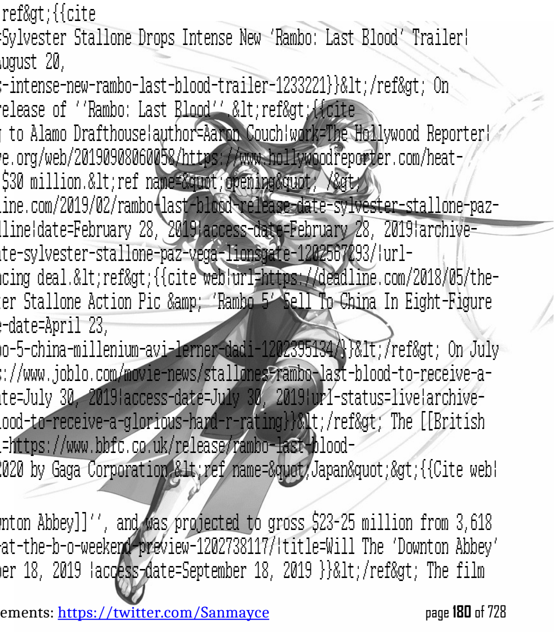
2667 &lt;ref name=&quot;OldTeam&quot;&gt;{{Cite web |last=Daniell |first=Mark |date=April 25, 2017 |title=Sylvester Stallone's 'Guardians of the Galaxy Vol. 2' character has a big future in the MCU |url=https://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu |url-status=live |archive-url=https://web.archive.org/web/20170428051803/https://www.thewhig.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu |archive-date=April 28, 2017 |access-date=April 28, 2017 |website=The Whig}}&lt;/ref&gt;

2668 | 2006-08 || ''[[Star Stories]]'' || [[Simon Fuller]], [[George Michael]]'s father, [[Bono]], [[David Jason]], [[Vinnie Jones]], [[Sylvester Stallone]], [[Ray Winstone]], [[George Clooney]], [[Matt LeBlanc]], [[Jason Orange]], [[Gary Barlow]], [[Rick Astley]], [[Jerome Flynn]], [[Dave Lee Travis]], [[Graham Taylor]], Mike Reid, [[Dane Bowers]], [[Mark Wahlberg|Marky Mark]] || 15 episodes

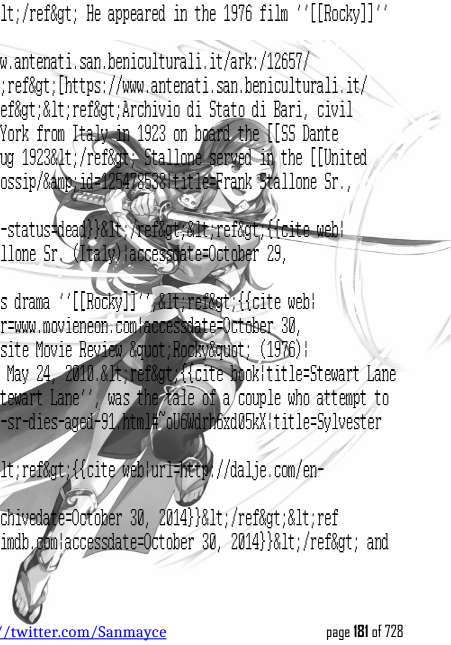


```
2669 * [[Sylvester Stallone]] (1986)
2670 ¦[[Sylvester Stallone]]
2671 ¦The production budget of this PG-13-rated, now-likely conclusion to the [[Sylvester Stallone]]-directed 80s action nostalgia series is still unknown, but it is unlikely to be less than the $49 million it has so far made worldwide. Even so, it garnered substantially more Wiki views than actually successful films such as ''[[If I Stay (film)¦If I Stay]]'' (#164) and ''[[Let's Be Cops]]'' (#503)
2672 A seventh channel '''M-Net Movies Showcase''' was later dropped by M-Net on 1 April 2017 with the aim of creating limited run pop-up channels, airing films ranging from those starring [[Tom Cruise]], [[Sylvester Stallone]] and [[Will Smith]]. On 1 June 2017 it was replaced by Sundance TV. On 1 December 2017, M-Net Movies Zone was made available in HD to all households who owned an HD capable decoder such as the DStv Explora, HD Decoder, HD PVR and were on the all DStv packages. M-Net Movies Zone is the only channel airing on the [[GOtv Africa¦GOtv]] service via its Max and Plus packages.
2673 ''[[Block Mania (board game)¦Block Mania]]'' (1987), a wargame set in Mega City 1, was based on the Judge Dredd &quot;[[Block War]]&quot; storyline. Players represent the [[Militia¦Citi-Defence]] forces of one of two Housing Blocks (named after historical figures [[Samantha Fox¦Sammy Fox]] and [[Buddy Holly]]). Each Block is fighting a &quot;Block War&quot; against neighboring rival Housing Blocks using their arsenal of advanced weapons. Successful attacks penetrate armor or damage interior spaces and reduce the Block's structural integrity. The winner is the last Block left intact. This was expanded with additional rules and game pieces from the ''[[Mega-Mania]]'' box set (1987) (which added the [[Sylvester Stallone¦Sly Stallone]], and [[Richard M. Nixon¦Richard Nixon]] Blocks) and the ''Happy Hour'' article featured in ''White Dwarf'' #94.
2674 * [[Sylvester Stallone]]
2675 * Sylvester Stallone
2676 * Sylvester Stallone
2677 '''''Rambo: Last Blood'''''{{efn¦Also known as '''''Rambo V'''''&lt;ref&gt;{{cite web¦url=https://deadline.com/2018/05/sylvester-stallone-cannes-film-festival-rambo-v-1202621832/¦title=Sylvester Stallone Celebrated At Cannes: Promises 'Serious Vengeance' In 'Rambo V', Rebooting 'Cobra' &amp; Plans For New 'Rocky'¦first=Anthony¦last=D'Alessandro¦work=Deadline Hollywood¦date=May 24, 2019¦access-date=July 1, 2019¦url-status=live¦archive-date=July 2, 2019¦archive-url=https://web.archive.org/web/20190702024917/https://deadline.com/2019/05/sylvester-stallone-cannes-film-festival-rambo-v-1202621832/}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=https://news.yahoo.com/stallone-jokes-elocution-school-sneak-peek-rambo-v-162458911.html¦title=Stallone jokes about elocution school at sneak peek of 'Rambo V'¦first1= Fiachra¦last1=Gibbons¦first2=Nicolas¦last2=Pratviel¦work=Yahoo News¦date=May 24, 2019¦access-date=July 1, 2019¦url-status=live¦archive-date=July 2, 2019¦archive-url=https://web.archive.org/web/20190702024741/https://news.yahoo.com/stallone-jokes-elocution-school-sneak-peek-rambo-v-162458911.html}}&lt;/ref&gt; and '''''Rambo V: Last Blood'''''.&lt;ref&gt;{{cite web¦url=https://www.hollywoodreporter.com/heat-vision/rambo-v-last-blood-trailer-sylvester-stallone-faces-his-past-watch-1214376¦title=John Rambo Faces His Past in First Trailer For 'Last Blood'¦first=Ryan¦last=Parker¦work=The Hollywood Reporter¦date=May 30, 2019¦access-date=July 21, 2019}}&lt;/ref&gt;}} is a 2019 American [[Vigilante film¦vigilante]] [[action film]] directed by [[Adrian Grünberg]]. The screenplay was co-written by Matthew Cirulnick and [[Sylvester Stallone]], from a story by [[Dan Gordon (screenwriter)¦Dan Gordon]] and Stallone, and is based on the character [[John Rambo]] created by the author [[David Morrell]] for his novel ''[[First Blood (novel)¦First Blood]]''. A sequel to ''[[Rambo (2008 film)¦Rambo]]'' (2008), it is the fifth&lt;!-- DO NOT ADD &quot;AND FINAL INSTALLMENT&quot; HERE. SYLVESTER STALLONE SAID IF &quot;LAST BLOOD&quot; IS SUCCESSFUL, RAMBO 6 WILL HAPPEN. --&gt; installment in the [[Rambo (franchise)¦''Rambo'' franchise]] and stars Stallone as Rambo, alongside [[Paz Vega]], [[Sergio Peris-Mencheta]], [[Adriana Barraza]], [[Yvette Monreal]], Genie Kim aka Yenah Han, [[Joaquín Cosío]], and [[Óscar Jaenada]]. In the film, Rambo travels to [[Mexico]] to save his adopted niece, who has been kidnapped by a Mexican cartel and forced into prostitution.
2678 * [[Sylvester Stallone]] as [[John Rambo]]
2679 In February 2008, Sylvester Stallone revealed that making a fifth film would depend on the success of the fourth film, stating he was &quot;gearing one up&quot; and that it would &quot;be quite different&quot;.&lt;ref&gt;{{cite news¦url=https://www.reuters.com/article/latestCrisis/idUSL02468068 ¦title=Interview - Stallone challenges Myanmar junta, eyes Rambo 5¦access-date=October 4, 2018¦first=Michael¦last=Winfrey¦publisher=Reuters ¦ date=February 2, 2008}}&lt;/ref&gt; In March 2008, Stallone revealed he was &quot;half-way through&quot; writing ''Rambo V'', stating that it would not be another war movie, with Bulgaria being considered to double as Rambo's home in Arizona.&lt;ref&gt;{{cite web ¦url=http://www.moviehole.net/news/20080310_stallone_halfway_through_writi.html ¦title=Stallone &quot;half-way through&quot; writing Rambo 5 ¦access-date=October 4, 2018 ¦publisher=Moviehole.net ¦archive-url=https://web.archive.org/web/20080313155558/http://www.moviehole.net/news/20080310_stallone_halfway_through_writi.html ¦archive-date=March 13, 2008 ¦url-status=dead }}&lt;/ref&gt;&lt;ref&gt;{{cite web ¦url=http://www.moviehole.net/news/20080320_rambo_5_to_film_in_bulgaria.html ¦title=Rambo 5 to film in Bulgaria ¦access-date=May 9, 2008 ¦publisher=Moviehole.net ¦archive-url=https://web.archive.org/web/20080323173248/http://www.moviehole.net/news/20080320_rambo_5_to_film_in_bulgaria.html ¦archive-date=March 23, 2008 ¦url-status=dead }}&lt;/ref&gt;&lt;ref&gt;{{cite web ¦url=http://www.moviehole.net/news/20080322_rambo_will_return_to_america.html ¦title=Rambo will return to America! ¦access-date=May 9, 2008 ¦publisher=Moviehole.net ¦archive-url=https://web.archive.org/web/20080326131659/http://www.moviehole.net/news/20080322_rambo_will_return_to_america.html ¦url-status=dead ¦archive-date=March 26, 2008 }}&lt;/ref&gt; In February 2009, Stallone revealed that he was proceeding with the fifth film, but stated, &quot;The conflict is whether to do it in America or a foreign country.&quot;&lt;ref&gt;{{cite web¦url=https://uk.ign.com/articles/2009/02/02/stallone-plans-rambo-5¦title=Stallone Plans Rambo 5¦first=Orlando¦last=Parfitt¦work=IGN¦date=February 2, 2009¦access-date=October 4, 2018}}&lt;/ref&gt;
2680 In May 2010, Stallone revealed he was &quot;done&quot; with the character, stating, &quot;I think Rambo's pretty well done. I don't think there'll be any more. I'm about 99% sure I was going to do it ... but I feel that with Rocky Balboa, that character came complete circle. He went home. But for Rambo to go on another adventure might be, I think, misinterpreted as a mercenary gesture and not necessary. I don't want that to happen.&quot;&lt;ref&gt;{{cite web¦url=https://www.cbr.com/sylvester-stallone-retires-rambo-wont-pursue-fifth-installment/¦title=Sylvester Stallone Retires Rambo, Won't Pursue Fifth Installment¦first=Josh¦last=Wigler¦work=CBR¦date=May 3, 2010¦access-date=October 4, 2018}}&lt;/ref&gt; At the [[2010 Cannes Film Festival]], Millennium Films and Nu Image advertised the film, now titled ''Rambo V'', with handout poster cards.&lt;ref&gt;{{cite web ¦access-date=May 26, 2019 ¦archive-date=October 12, 2018 ¦archive-url=https://web.archive.org/web/20181012053938/http://www.worstpreviews.com/headline.php?id=17692&amp;count=0 ¦title=Why is ¦url=http://www.worstpreviews.com/headline.php?id=17692&amp;count=0 ¦url-status=dead ¦website=worstpreviews.com}}&lt;/ref&gt; Despite this, Stallone stated that he wasn't involved with the created posters and that the studio had told him that if he doesn't make the film they'll find someone else to complete it. He stated he wanted the franchise to end, and expressed being comfortable with the idea.&lt;ref&gt;{{cite web¦url=http://legacy.aintitcool.com/node/45076¦title=So Sylvester Stallone just called¦last=headgeek¦website=Aint It Cool News¦access-date=May 26, 2019}}&lt;/ref&gt;
2681 {{Anchor¦Rambo: Last Stand¦Last Stand}}In 2011, [[Sean Hood]] was hired to write a new script titled ''Rambo: Last Stand'', which he described as &quot;more in line with the small-town thriller of ''[[First Blood]]''&quot;.&lt;ref&gt;{{cite web¦url=https://www.empireonline.com/movies/news/new-screenplay-rambo-5/¦title=New Screenplay For Rambo 5¦first=Owen¦last=Williams¦work=Empire¦date=August 19, 2011¦access-date=October 2, 2018¦url-status=live¦archive-date=April 2, 2019¦archive-url=https://web.archive.org/web/20190402011517/https://www.empireonline.com/movies/news/new-screenplay-rambo-5/}}&lt;/ref&gt; In 2012, Hood revealed that ''Rambo V'' had been put on hold in order for Stallone to finish ''[[The Expendables 2]]''. Hood also revealed his uncertainty on whether the film would be similar to ''[[Unforgiven]]'' or a &quot;passing-of-the-torch&quot;.&lt;ref&gt;{{cite web¦url=https://www.joblo.com/horror-movies/news/sean-hood-finally-gives-an-update-on-the-state-of-rambo-5¦title=Sean Hood Finally Gives an Update on the State of Rambo 5¦first=Marcey¦last=Papandrea¦work=Joblo¦date=February 10, 2012¦access-date=October 2, 2018}}&lt;/ref&gt; In August 2013, it was announced that [[Entertainment One]] and Nu Image would develop and produce a ''Rambo'' TV series with Stallone.&lt;ref&gt;{{cite web¦url=https://www.hollywoodreporter.com/news/sylvester-stallone-talks-rambo-tv-611150¦title=Sylvester Stallone In Talks For 'Rambo' TV Series¦first=Etan¦last=Vlessing¦work=The Hollywood Reporter¦date=August 21, 2013¦access-date=October 4, 2018}}&lt;/ref&gt; In June 2014, German film company Splendid Films confirmed that Stallone had started writing the script for ''Rambo V'', which he described as similar to ''[[No Country for Old Men]]''.&lt;ref&gt;{{cite web¦url=https://theplaylist.net/no-country-for-old-men-esque-rambo-v-reportedly-on-the-way-20140623/¦title='No Country for Old Men'-Esque 'Rambo V' Reportedly On The Way¦first=Kevin¦last=Jagernauth¦work=The Playlist ¦date=June 23, 2014¦access-date=August 27, 2021¦url-status=live¦archive-date=August 28, 2021¦archive-url=https://web.archive.org/web/20210828030403/https://theplaylist.net/no-country-for-old-men-esque-rambo-v-reportedly-on-the-way-20140623/}}&lt;/ref&gt; In September 2014, the film was officially titled ''Rambo: Last Blood'', with Stallone also serving as director.&lt;ref&gt;{{cite web¦url=https://www.comingsoon.net/movies/news/122801-sylvester-stallone-targets-one-last-battle-with-rambo-last-blood¦title=Sylvester Stallone Targets One Last Battle with Rambo: Last Blood¦first=Silas¦last=Lesnick¦work=Coming Soon¦date=September 10, 2014¦access-date=October 4, 2018}}&lt;/ref&gt;
2682 In 2015, Stallone and Rambo creator/author [[David Morrell]] re-wrote the story for ''Rambo V''. Stallone wanted a &quot;soulful journey&quot; for the character that the author described as a &quot;really emotional, powerful story&quot;. Stallone pitched the idea to the producers, but they wanted to proceed with the human trafficking story instead, leaving Stallone and Morrell's script abandoned.&lt;ref&gt;{{cite web¦url=https://www.culturecreature.com/david-morrell-interview-rambo/¦title='Rambo' Creator David Morrell Discusses His Journey With A Cinematic Icon¦first=Dan¦last=Redding¦work=Culture Creature¦date=October 29, 2018¦access-date=October 29, 2018}}&lt;/ref&gt; In October 2015, Stallone expressed interest in a prequel project, stating: &quot;It's intriguing to find the whys and wherefores of how people have become what they are. The traumas, the loss, and the tragedy of being in Vietnam would certainly be a great challenge for a young actor, and it would be ironic that Rambo directs younger Rambo having played it for twenty years plus.&quot;&lt;ref&gt;{{cite web¦url=https://www.empireonline.com/movies/news/stallone-ponders-rambo-prequel/¦title=Stallone Ponders Rambo Prequel¦first=Owen¦last=Williams¦work=Empire¦date=October 8, 2015¦access-date=October 4, 2018}}&lt;/ref&gt; By January 2016, Stallone stated that ''Rambo V'' would not enter production.&lt;ref&gt;{{cite web¦work=[[Variety (magazine)¦Variety]]¦url=https://variety.com/2016/film/news/sylvester-stallone-retiring-rambo-1201672291/¦title=Sylvester Stallone Is Retiring From Playing Rambo¦author=Setoodeh, Ramin¦date=January 5, 2016}}&lt;/ref&gt;
2683 In May 2018, the film was re-announced to be in development initially as ''Rambo V'', with principal photography scheduled for September. The plot was confirmed to focus on Rambo combating a Mexican drug cartel following their abduction of one of his family members.&lt;ref name=&quot;ScreenDaily&quot;&gt;{{cite news¦last=Kay¦first=Jeremy¦url=https://www.screendaily.com/sylvester-stallone-lining-up-rambo-v-exclusive/5128882.article¦title=Sylvester Stallone lining up 'Rambo V' (exclusive)¦publisher=[[Screen Daily]]¦date=May 5, 2018¦access-date=May 8, 2018}}&lt;/ref&gt; Stallone announced the release date for the film, which he would also be co-writing with Matthew Cirulnick. At that time, he expressed uncertainty in additionally serving as director.&lt;ref name=&quot;Cannes May 2018&quot;&gt;{{cite web¦last=Wiseman¦first=Andrew¦work=[[Deadline Hollywood¦Deadline]]¦url=https://deadline.com/2018/05/sylvester-stallone-in-line-to-take-on-mexican-cartel-in-rambo-5-1202383468/¦title=Sylvester Stallone In Line To Take on Mexican Cartel in Rambo 5¦date=May 5, 2018¦access-date=May 7, 2018¦url-status=live¦archive-date=February 14, 2019¦archive-url=https://web.archive.org/web/20190214104758/https://deadline.com/2018/05/sylvester-stallone-in-line-to-take-on-mexican-cartel-in-rambo-5-1202383468/}}&lt;/ref&gt;&lt;ref name=&quot;IGN&quot;&gt;{{cite news¦last=Gilyadov¦first=Alex¦url=https://www.ign.com/articles/2018/05/08/rambo-5-reportedly-in-the-works-with-sylvester-stallone¦title=Rambo 5 Confirmed by Stallone, Coming Fall 2019¦website=IGN¦date=May 8, 2018¦access-date=May 8, 2018¦archive-url=https://web.archive.org/web/20180509082229/https://www.ign.com/articles/2018/05/08/rambo-5-reportedly-in-the-works-with-sylvester-stallone¦archive-date=May 9, 2018}}&lt;/ref&gt; In August 2018, [[Adrian Grünberg]] was announced as the director.&lt;ref&gt;{{cite web¦url=https://screenrant.com/rambo-5-director/¦title=Rambo 5 Reportedly Recruits Get the Gringo Director¦first=David¦last=Stephens¦work=Screen Rant¦date=August 18, 2018¦access-date=October 2, 2018}}&lt;/ref&gt; In September 2018, [[Adriana Barraza]] was added to the cast.&lt;ref name=&quot;Barraza&quot;&gt;{{cite web¦url=http://esdiario.com.mx/?p=215591¦title=Adriana Barraza se prepara para Rambo V¦website=Esdiario.com.mx¦access-date=May 26, 2019¦archive-date=April 2, 2019¦archive-url=https://web.archive.org/web/20190402030608/http://esdiario.com.mx/?p=215591¦url-status=dead}}&lt;/ref&gt; In October 2018, [[Paz Vega]],&lt;ref name=&quot;vega&quot;&gt;{{cite web¦last=Mia¦first=Galuppo¦work=[[The Hollywood Reporter]]¦url=https://www.hollywoodreporter.com/heat-vision/rambo-5-paz-vega-star-opposite-sylvester-stallone-1149003¦title='Rambo 5': Paz Vega to Star Opposite Sylvester Stallone¦date=October 8, 2018¦access-date=October 8, 2018}}&lt;/ref&gt; [[Yvette Monreal]],&lt;ref
```

```
     name=&quot;MonrealCast&quot;&gt;{{cite news|last=Galuppo|first=Mia|title=Sylvester Stallone's 'Rambo 5' Adds Yvette Monreal (Exclusive)|url=https://www.hollywoodreporter.com/heat-vision/rambo-5-
     casts-yvette-monreal-1151099|access-date=October 10, 2018|work=[[The Hollywood Reporter]]|date=October 10, 2018|language=en}}&lt;/ref&gt; [[Sergio Peris-Mencheta]],&lt;ref&gt;{{cite
     web|url=https://deadline.com/2018/10/rambo-5-sergio-peris-mencheta-cast-villain-1202480258/|title='Rambo 5' Finds Its Bad Guy In 'Snowfall's Sergio Peris-Mencheta|first=Patrick|last=Hipes|
     work=[[Deadline Hollywood|Deadline]]|date=October 10, 2018|access-date=October 11, 2018}}&lt;/ref&gt; [[Óscar Jaenada]], and [[Joaquín Cosío]]&lt;ref name=deadlineoct23&gt;{{cite
     web|url=https://deadline.com/2018/10/sylvester-stallone-rambo-last-blood-oscar-jaenada-1202488090/|title=Sylvester Stallone's 'Rambo: Last Blood' Adds Oscar Jaenada|work=Deadline|first=Amanda|
     last=N'Duka|date=October 23, 2018|access-date=October 28, 2018}}&lt;/ref&gt; were cast. In May 2019, [[Louis Mandylor]], Sheila Shah, [[Dimitri Vegas &amp; Like Mike|Dimitri Vegas]], and Genie Kim
     (aka Yenah Han) were revealed to have been cast without prior announcement.&lt;ref name=&quot;cast&quot;&gt;{{cite web|url=https://collider.com/rambo-5-trailer/|title='Rambo: Last Blood': First
     Trailer Reveals John Rambo's Last Ride|first=Dave|last=Trumbore|date=May 30, 2019|website=Collider.com|access-date=May 30, 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite
     web|url=https://www.geekalerts.com/rambo-last-blood-teaser-trailer/|title=Rambo: Last Blood - Teaser Trailer|first=Ryan|last=Cooper|work=Geek Alerts|date=May 30, 2019|access-date=May 30, 2019|url-
     status=live|archive-date=May 30, 2019|archive-url=https://web.archive.org/web/20190530220856/https://www.geekalerts.com/rambo-last-blood-teaser-trailer/}}&lt;/ref&gt;&lt;ref&gt;{{cite web|
     url=https://variety.com/2019/music/news/dimitri-vegas-rambo-v-last-blood-sylvester-stallone-1203219408/|title=Dimitri Vegas Talks 'Rambo V: Last Blood' Role, Working With Sylvester Stallone|
     first=Shirley|last=Halperin|work=Variety|date=May 19, 2019|access-date=June 1, 2019|url-status=live|archive-date=June 1, 2019|archive-url=https://web.archive.org/web/20190601212919/https://
     variety.com/2019/music/news/dimitri-vegas-rambo-v-last-blood-sylvester-stallone-1203219408/}}&lt;/ref&gt;
2684 [[Principal photography]] began on October 2, 2018 in Bulgaria.&lt;ref&gt;{{cite news|last=Alan Orange|first=B.|title=Stallone Returns in First Rambo 5 Set Photos as a Badass
     Cowboy|url=https://movieweb.com/rambo-5-set-photos-sylvester-stallone-cowboy/|access-date=October 3, 2018|work=MovieWeb|date=October 2, 2018}}&lt;/ref&gt;&lt;ref&gt;{{cite news|last=N'Duka|
     first=Amanda|title='Rambo': Sylvester Stallone Offers First Look At The Fifth Installment|url=https://deadline.com/2018/10/rambo-sylvester-stallone-offer-first-look-photos-rambo-5-1202475306/|access-
     date=October 3, 2018|work=[[Deadline Hollywood|Deadline]]|date=October 3, 2018}}&lt;/ref&gt; It was previously scheduled to begin on September 1, 2018,&lt;ref&gt;{{cite news|last=Kay|first=Jeremy|
     title=Sylvester Stallone lining up 'Rambo V' (exclusive)|url=https://www.screendaily.com/sylvester-stallone-lining-up-rambo-v-exclusive/5128882.article|access-date=May 7, 2018|work=Screen|date=May 6,
     2018|language=en}}&lt;/ref&gt;&lt;ref&gt;{{cite news|last=Wiseman|first=Andreas|title=Sylvester Stallone In Line To Take On Mexican Cartel In 'Rambo 5'|url=https://deadline.com/2018/05/sylvester-
     stallone-in-line-to-take-on-mexican-cartel-in-rambo-5-1202383468/|access-date=May 29, 2018|work=[[Deadline Hollywood|Deadline]]|date=May 5, 2018}}&lt;/ref&gt; and before that on October 27, 2014, in
     [[Shreveport, Louisiana|Shreveport]], [[Louisiana]].&lt;ref&gt;{{cite news|last=Scott|first=Mike|title=Sylvester Stallone's 'Rambo 5' to shoot in Louisiana next
     month|url=http://www.nola.com/movies/index.ssf/2014/09/sylvester_stallones_rambo_5_to.html|access-date=September 16, 2014|publisher=nola.com|date=September 8, 2014}}&lt;/ref&gt;&lt;ref&gt;{{cite
     news|last=A.|first=Jonathan|title='Rambo 5″ Starring Sylvester Stallone Now Hiring Crew Members in Louisiana|url=http://www.projectcasting.com/casting-calls-and-auditions/rambo-5-starring-sylvester-
     stallone-now-hiring-crew-members-in-louisiana/|access-date=October 28, 2014|publisher=projectcasting.com|date=September 12, 2014|archive-date=November 2,
     2014|archive-url=https://web.archive.org/web/20141102004430/http://www.projectcasting.com/casting-calls-and-auditions/rambo-5-starring-sylvester-stallone-now-hiring-crew-members-in-louisiana/|url-
     status=dead}}&lt;/ref&gt; Barraza filmed her scenes in [[Tenerife]] ([[Canary Islands]]).&lt;ref name=&quot;Barraza&quot;/&gt; Principal photography was completed on December 4, 2018.&lt;ref&gt;
     {{cite web|url=https://screenrant.com/sylvester-stallone-rambo-5-wrap-filming/|title=Rambo 5: Sylvester Stallone Wraps Filming|first=Padraig|last=Cotter|work=Screen Rant|date=December 4, 2018|access-
     date=December 9, 2018}}&lt;/ref&gt; Additional photography took place at the end of May 2019.&lt;ref&gt;{{Cite web|url=https://kanal3.bg/news/98093-Staloun-razmaha-za-posledno-yumruci-v-Kan-predi-
     +balgarskiya+-Rambo-V%253A-Posledna-krav-%2528SNIMKI%2529|title=Сталоун размаха за последно юмруци в Кан преди|website=kanal3.bg|access-date=December 8, 2020|archive-date=May 26, 2019|archive-
     url=https://web.archive.org/web/20190526115936/https://kanal3.bg/news/98093-Staloun-razmaha-za-posledno-yumruci-v-Kan-predi-+balgarskiya+-Rambo-V%253A-Posledna-krav-%2528SNIMKI%2529|url-
     status=dead}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.24chasa.bg/ojivlenie/article/7469992|title=Силвестър Сталоун идва в България, за да презаснеме &quot;Рамбо V&quot;|
     website=www.24chasa.bg|date=May 25, 2019 }}&lt;/ref&gt;
2685 In May 2018, Millennium Films brought the project to Cannes to generate interest and sales.&lt;ref name=&quot;Cannes May 2018&quot;/&gt; Stallone verified that he would share images and videos from
     the film's set on his Instagram as the film approached its release.&lt;ref&gt;{{cite web|url=https://screenrant.com/rambo-5-last-blood-release-date-trailer-story/|title=Everything You Need To Know
     About Rambo V: Last Blood|first=Padraig|last=Cotter|work=Screen Rant|date=January 22, 2019|access-date=February 14, 2019|archive-date=February 14,
     2019|archive-url=https://web.archive.org/web/20190214103601/https://screenrant.com/rambo-5-last-blood-release-date-trailer-story/}}&lt;/ref&gt;&lt;ref&gt;{{cite news |title=Sylvester Stallone tries
     to 'escape' camera crew on Rambo 5 film set|url=https://filmindustry.network/sylvester-stallone-tries-to-escape-camera-crew-on-rambo-5-film-set/34143 |publisher=Film Industry Network |date=March 3,
     2019}}&lt;/ref&gt; In February 2019, Stallone revealed images on his Instagram of Rambo's adopted family,&lt;ref&gt;{{cite web|url=https://wegotthiscovered.com/movies/rambo-v-last-blood-set-photos-
     show-sylvester-stallone-opposite-new-costars/|title=Rambo V: Last Blood Set Pics Reveal Stallone's New Co-Stars|first=David|last=Pountain|work=We Got This Covered|date=February 4, 2019|access-
     date=February 14, 2019|archive-date=February 14, 2019|archive-url=https://web.archive.org/web/20190214104024/https://wegotthiscovered.com/movies/rambo-v-last-blood-set-photos-show-sylvester-stallone-
     opposite-new-costars/}}&lt;/ref&gt; combat history,&lt;ref&gt;{{cite web|url=https://screenrant.com/rambo-5-photos-sylvester-stallone-backstory/|title=New Rambo V: Last Blood Photos Reveal His Combat
     History|first=Padraig|last=Cotter|work=Screen Rant|date=February 12, 2019|access-date=February 14, 2019|url-status=live|archive-date=February 14,
     2019|archive-url=https://web.archive.org/web/20190214104226/https://screenrant.com/rambo-5-photos-sylvester-stallone-backstory/}}&lt;/ref&gt; and the character Gabriela's intentions to travel to
     Mexico to find her father.&lt;ref name=&quot;Feb 22&quot;&gt;{{cite web|url=https://screenrant.com/rambo-5-last-blood-photos-stallone-plot-details/|title=New Rambo V: Last Blood Images Set Up Rescue
     Mission Storyline|first=Padraig|last=Cotter|work=Screen Rant|date=February 22, 2019|access-date=February 22, 2019}}&lt;/ref&gt; In March 2019, Stallone revealed via his Instagram an image of Rambo
     covered in blood and aiming his signature bow.&lt;ref&gt;{{cite web|url=https://www.cinemablend.com/news/2468309/sylvester-stallone-is-bloody-and-packing-heat-in-new-rambo-last-blood-image|
     title=Sylvester Stallone Is Bloody And Packing Heat In New Rambo: Last Blood Image|first=Dirk|last=Libbey|work=Cinema Blend|date=March 13, 2019|access-date=March 15, 2019}}&lt;/ref&gt;
2686 In May 2019, it was revealed that Stallone will present exclusive images at Cannes to coincide with a special &quot;first-look&quot; screening of the film at the [[Palais des Festivals et des
     Congrès]] on May 24, 2019.&lt;ref&gt;{{cite web|url=https://variety.com/2019/film/news/sylvester-stallone-rambo-v-cannes-1203209019/|title=Sylvester Stallone to Share First Look at 'Rambo V' at
     Cannes Film Festival|first=Jordan|last=Moreau|work=Variety|date=May 8, 2019|access-date=May 9, 2019|url-status=live|archive-date=May 9,
     2019|archive-url=https://web.archive.org/web/20190509075932/https://variety.com/2019/film/news/sylvester-stallone-rambo-v-cannes-1203209019/}}&lt;/ref&gt; The first trailer was revealed at Cannes on
     May 24, 2019.&lt;ref&gt;{{cite web|url=https://deadline.com/2019/05/rambo-v-trailer-sylvester-stallone-cannes-tribute-1202622056/|title=Sylvester Stallone Unveils 'Rambo V' Trailer At Cannes Tribute
     &amp; Remembers How 'First Blood' Changed His Life|first=Anthony|last=D'Alessandro|work=Deadline|date=May 24, 2019|access-date=May 26, 2019|url-status=live|archive-date=May 26, 2019|archive-
     url=https://web.archive.org/web/20190526195445/https://deadline.com/2019/05/rambo-v-trailer-sylvester-stallone-cannes-tribute-1202622056/}}&lt;/ref&gt; The teaser trailer was released on May 30,
     2019&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/heat-vision/rambo-v-last-blood-trailer-sylvester-stallone-faces-his-past-watch-1214376|title=John Rambo Faces His Past in First Trailer
     For 'Last Blood'|first=Ryan|last=Parker|work=The Hollywood Reporter|date=May 30, 2019|access-date=May 30, 2019|url-status=live|archive-date=May 30,
     2019|archive-url=https://web.archive.org/web/20190530194842/https://www.hollywoodreporter.com/heat-vision/rambo-v-last-blood-trailer-sylvester-stallone-faces-his-past-watch-1214376}}&lt;/ref&gt; and
     drew comparisons to ''[[Logan (film)|Logan]]'' and ''[[Unforgiven]]''.&lt;ref&gt;{{Cite web|url=https://www.comicsbeat.com/rambo-returns-in-rambo-last-blood-trailer/|title=Syndicated Comics|
     first=Hussein|last=Wasiti|date=May 31, 2019}}&lt;/ref&gt; A remixed version of the song &quot;[[Old Town Road]]&quot; was used for the teaser.&lt;ref&gt;{{cite
     web|url=https://www.thefader.com/2019/05/30/rambo-last-blood-trailer-old-town-road-mass-murder|title=The Rambo: Last Blood trailer features &quot;Old Town Road,&quot; mass murder|first=Jordan|
     last=Darville|work=The Fader|date=May 30, 2019|access-date=August 11, 2019|url-status=live|archive-date=August 11, 2019|archive-url=https://web.archive.org/web/20190811215017/https://
     www.thefader.com/2019/05/30/rambo-last-blood-trailer-old-town-road-mass-murder}}&lt;/ref&gt; On August 1, 2019, Stallone revealed the theatrical release poster on his Instagram.&lt;ref&gt;{{cite web|
     url=https://www.comingsoon.net/movies/news/1088265-check-out-a-new-poster-for-rambo-last-blood|title=New Poster for Rambo: Last Blood Flexes Its Muscles|first=Jeff|last=Ames|work=Comingsoon.net|
     date=August 1, 2019|access-date=August 2, 2019|url-status=live|archive-date=August 2, 2019|archive-url=https://web.archive.org/web/20190802090244/https://www.comingsoon.net/movies/news/1088265-check-
     out-a-new-poster-for-rambo-last-blood}}&lt;/ref&gt; On August 20, 2019, Stallone released the second trailer on his Instagram.&lt;ref&gt;{{cite
     web|url=https://www.hollywoodreporter.com/heat-vision/sylvester-stallone-drops-intense-new-rambo-last-blood-trailer-1233221|title=Sylvester Stallone Drops Intense New 'Rambo: Last Blood' Trailer|
     first=Ryan|last=Parker|work=The Hollywood Reporter|date=August 20, 2019|access-date=August 21, 2019|url-status=live|archive-date=August 20,
     2019|archive-url=https://web.archive.org/web/20190820183843/https://www.hollywoodreporter.com/heat-vision/sylvester-stallone-drops-intense-new-rambo-last-blood-trailer-1233221}}&lt;/ref&gt; On
     September 4, 2019, [[Alamo Drafthouse Cinema]] announced it would host a marathon of all five ''Rambo'' films to commemorate the release of ''Rambo: Last Blood''.&lt;ref&gt;{{cite
     web|url=https://www.hollywoodreporter.com/heat-vision/rambo-marathon-coming-alamo-drafthouse-1236671|title='Rambo' Marathon Coming to Alamo Drafthouse|author=Aaron Couch|work=The Hollywood Reporter|
     date=September 4, 2019|access-date=September 7, 2019|url-status=live|archive-date=September 8, 2019|archive-url=https://web.archive.org/web/20190908060058/https://www.hollywoodreporter.com/heat-
     vision/rambo-marathon-coming-alamo-drafthouse-1236671}}&lt;/ref&gt; [[Deadline Hollywood]] reported that P&amp;A costs were under $30 million.&lt;ref name=&quot;opening&quot; /&gt;
2687 ''Rambo: Last Blood'' was released theatrically in the United States on September 20, 2019.&lt;ref&gt;{{cite web|url=https://deadline.com/2019/02/rambo-last-blood-release-date-sylvester-stallone-paz-
     vega-lionsgate-1202567293/|title='Rambo: Last Blood' Will Be Spilled In September Via Lionsgate|first=Erik|last=Pedersen|work=Deadline|date=February 28, 2019|access-date=February 28, 2019|archive-
     date=March 1, 2019|archive-url=https://web.archive.org/web/20190301041816/https://deadline.com/2019/02/rambo-last-blood-release-date-sylvester-stallone-paz-vega-lionsgate-1202567293/|url-
     status=live}}&lt;/ref&gt; Dadi purchased the [[Mainland China|Chinese]] distribution rights and agreed to an eight-figure co-financing deal.&lt;ref&gt;{{cite web|url=https://deadline.com/2018/05/the-
     expendables-4-sylvester-stallone-rambo-5-china-millenium-avi-lerner-dadi-1202395134/|title='The Expendables 4' Muscles Up: Sylvester Stallone Action Pic &amp; 'Rambo 5' Sell To China In Eight-Figure
     Co-Fi Deal|first=Andreas|last=Wiseman|work=Deadline Hollywood|date=May 21, 2018|access-date=April 22, 2019|url-status=live|archive-date=April 23,
     2019|archive-url=https://web.archive.org/web/20190423013848/https://deadline.com/2018/05/the-expendables-4-sylvester-stallone-rambo-5-china-millenium-avi-lerner-dadi-1202395134/}}&lt;/ref&gt; On July
     30, 2019, the [[Motion Picture Association film rating system|MPAA]] assigned the film an R rating.&lt;ref&gt;{{cite web|url=https://www.joblo.com/movie-news/stallones-rambo-last-blood-to-receive-a-
     glorious-hard-r-rating|title=Stallone's Rambo: Last Blood to receive a glorious hard R rating|first=Steve|last=Seigh|work=Joblo|date=July 30, 2019|access-date=July 30, 2019|url-status=live|archive-
     date=July 31, 2019|archive-url=https://web.archive.org/web/20190731052335/https://www.joblo.com/movie-news/stallones-rambo-last-blood-to-receive-a-glorious-hard-r-rating}}&lt;/ref&gt; The [[British
     Board of Film Classification]] classified the film at [[18 (British Board of Film Classification)|18]].&lt;ref&gt;{{cite web | url=https://www.bbfc.co.uk/release/rambo-last-blood-
     q29sbgvjdglvbjpwwc00otc2mzi | title=Rambo: Last Blood }}&lt;/ref&gt; The film was released theatrically in [[Japan]] on June 26, 2020 by Gaga Corporation.&lt;ref name=&quot;Japan&quot;&gt;{{Cite web|
     url=https://eiga.com/movie/91852/|title=ランボー ラスト・ブラッド|publisher=Eiga|access-date=July 27, 2020}}&lt;/ref&gt;
2688 In the United States and Canada, the film was released alongside ''[[Ad Astra (film)|Ad Astra]]'' and ''[[Downton Abbey (film)|Downton Abbey]]'', and was projected to gross $23-25 million from 3,618
     theaters in its opening weekend.&lt;ref&gt;{{Cite web|url=https://deadline.com/2019/09/will-the-downton-abbey-gang-take-out-rambo-at-the-b-o-weekend-preview-1202738117/|title=Will The 'Downton Abbey'
     Gang Take Out 'Rambo'? - Weekend Box Office Preview|last=D'Alessandro |first=Anthony |website=[[Deadline Hollywood]] |date=September 18, 2019 |access-date=September 18, 2019 }}&lt;/ref&gt; The film
```



```
made $7.17 million on Friday, which included $1.3 million from Thursday night previews.&lt;ref&gt;{{cite news|last=Mendelson|first=Scott|title=Box Office: Sylvester Stallone's 'Rambo' And Brad Pitt's 'Ad Astra' Both Earn $7 Million On Friday|url=https://www.forbes.com/sites/scottmendelson/2019/09/21/box-office-sylvester-stallones-rambo-and-brad-pitts-ad-astra-both-earn-7-million-on-friday/#302b6df132a3|access-date=September 21, 2019|work=[[Forbes]]|date=September 21, 2019}}&lt;/ref&gt; It went on to debut to $19 million, finishing third and marking the second-best opening of the series.&lt;ref name=&quot;opening&quot;&gt;{{Cite web|url=https://deadline.com/2019/09/downton-abbey-rambo-last-blood-ad-astra-hustlers-weekend-box-office-1202739525/|title= Big Screen Take On TV Series 'Downton Abbey' Is A Hit With $31M+ Opening, Reps Record For Focus Features - Early Sunday Update |last=D'Alessandro |first=Anthony |website=[[Deadline Hollywood]] |date=September 22, 2019 |access-date=September 22, 2019}}&lt;/ref&gt; The film made $8.6 million in its second weekend and $3.6 million in its third, finishing sixth and eighth, respectively.&lt;ref&gt;{{cite web |url=https://deadline.com/2019/09/abominable-downton-abbey-hustlers-weekend-box-office-1202746278/ |title=Dreamworks Animation-Pearl Studios' 'Abominable' Bigfoots B.O. With Near $21M Opening Weekend
The portrayal of the Rambo character was put under scrutiny. Writing for the ''Los Angeles Times'', [[Kenneth Turan]] gave the film 3 out of 4 stars and was complimentary of the story's &quot;surprisingly brooding examination&quot; of Rambo: &quot;Sure, Rambo is convincing when he ends up telling bad people, 'I'm gonna hurt you real bad', but there is also a kind of fragility that makes us worry about people putting the hurt on him.&quot;&lt;ref name=&quot;turan&quot;&gt;{{cite web|last=Turan|first=Kenneth|title=Review: Sylvester Stallone's 'Rambo: Last Blood' goes for the jugular - but in darker ways|url=https://www.latimes.com/entertainment-arts/movies/story/2019-09-19/rambo-last-blood-review-sylvester-stallone|website=Los Angeles Times|date=September 19, 2019|access-date=September 20, 2019}}&lt;/ref&gt; Peter Debruge wrote a negative review for ''[[Variety (magazine)|Variety]]'': &quot;This character is a mess of contradictions, representing, on one hand, the permanent damage that military service can do to one's soul while simultaneously suggesting what the ideal soldier looks like.&quot;&lt;ref name=&quot;debruge&quot;&gt;{{cite web|last=Debruge|first=Peter |title=Film Review: 'Rambo: Last Blood'|url=https://variety.com/2019/film/reviews/rambo-last-blood-review-sylvester-stallone-1203341831/ |website=Variety |date=September 19, 2019 |access-date=September 19, 2019}}&lt;/ref&gt; Rating the film 4.5 out of 10 for [[IGN]], Witney Seibold lamented, &quot;A character who was originally meant to stand as a symbol for the damage that war can do to a soldier is now best remembered as an unkillable human machine gun&quot;, but credited Stallone with &quot;[managing] to give as soulful a performance as the part warrants&quot;.&lt;ref name=&quot;seibold&quot;&gt;{{cite web|last=Seibold|first=Witney |title=Rambo: Last Blood Review|url=https://www.ign.com/articles/2019/09/18/rambo-last-blood-review |website=[[IGN]] |date=September 19, 2019 |access-date=September 19, 2019}}&lt;/ref&gt;
| [[Sylvester Stallone]]
| Sylvester Stallone
| Sylvester Stallone &amp; his impotent rage
| Matthew Cirulnick &amp; Sylvester Stallone
During Cannes 2019, Stallone said he would continue portraying Rambo if the fifth film succeeds.&lt;ref&gt;{{cite web|url=https://www.joblo.com/horror-movies/news/stallone-would-return-as-rambo-if-last-blood-is-a-hit|title= Stallone happy to return as Rambo if Last Blood is a hit|first=Mike|last=Sprague|date=June 12, 2019|publisher=[[JoBlo.com]]|access-date=June 13, 2019}}&lt;/ref&gt; Grünberg, however, said that ''Rambo: Last Blood'' &quot;closes the circle&quot;, hoping it would conclude the film series.&lt;ref name=&quot;Total Film&quot;&gt;{{cite web|url=https://www.gamesradar.com/rambo-5-last-blood-director-adrian-grunberg-interview/|title= Exclusive: Rambo: Last Blood director discusses bringing back Sylvester Stallone's action hero: &quot;This movie closes the circle&quot;|author=Jack Shepherd, James Mottram|work=[[GamesRadar]]|date=July 22, 2019|access-date=July 25, 2019}}&lt;/ref&gt; In September 2019, Stallone confirmed that he has plans for a prequel to the series; although he would not reprise the title role, he would like to explore who Rambo was before the [[Vietnam War]]: {{blockquote|I always thought of Rambo when he was 16 or 17-I hope they can do the prequel-he was the best person you could find. He was the captain of the team; he was the most popular kid in school; a super athlete. He was like [[Jim Thorpe]], and the war is what changed him. If you saw him before, he was like the perfect guy.&lt;ref&gt;{{cite web|last=Zinski|first=Dan |title=Sylvester Stallone Wants a Rambo Prequel|url=https://screenrant.com/rambo-prequel-movie-idea-sylvester-stallone/ |website=[[Screen Rant]] |date=September 16, 2019 |access-date=17 September 2019}}&lt;/ref&gt;|sign=|source=}}
In September 2019, Stallone had expressed interest in having Rambo take refuge in an [[Indian reservation]] for a potential sixth film.&lt;ref&gt;{{cite web|title=Sylvester Stallone interview for Rambo: Last Blood!|url=https://www.youtube.com/watch?v=2th50Z9nKcI|work=[[JoBlo.com]]| date=September 17, 2019 |publisher=JoBlo.com official YouTube channel|access-date=September 18, 2019|time=3:52}}&lt;/ref&gt; In June 2020, Stallone briefly elaborated on the idea, stating, &quot;If I ever did another one, I think he would go back to the Indian reservation that he grew up on because he has Indian family.&quot;&lt;ref&gt;{{Cite web|url=https://www.youtube.com/watch?v=otyZ9qS3-cY|title=ランボー ラスト・ブラッド シルベスター・スタローン主演 インタビュー最後の勇者編Rambo: Last Blood Sylvester Stallone interview movie|date=2020-06-23|work=[[Nippon TV]] Sukkiri|publisher=Shigz Channel, YouTube|access-date=June 25, 2020|time=8:30}}&lt;/ref&gt;
However, in a September 2023 convention appearance, Stallone cast doubt on a sixth film, joking that Rambo would just be fighting [[arthritis]].&lt;ref&gt;{{Cite web |last=Phillips |first=T. C. |date=2023-09-16 |title=&quot;What Am I Fighting? Arthritis?&quot; Rambo 6 Chances Get Honest Response From Sylvester Stallone |url=https://screenrant.com/rambo-6-movie-franchise-update-sylvester-stallone/ |access-date=2023-10-14 |website=ScreenRant |language=en}}&lt;/ref&gt;
{{Sylvester Stallone}}
[[Category:Films with screenplays by Sylvester Stallone]]
The film explores the effect of two Israeli cousins, [[Menahem Golan]] and [[Yoram Globus]], on Hollywood by producing films and starting one of the most successful independent production companies, [[The Cannon Group|Cannon Films]]. Apart from Golan and Globus, other Hollywood personalities appearing in the film include [[Sylvester Stallone]], [[Jon Voight]], [[Charles Bronson]], [[Chuck Norris]], [[Michael Dudikoff]], [[Billy Drago]], [[Andrey Konchalovskiy]] and [[Franco Zeffirelli]].&lt;ref&gt;{{cite web|url=http://www.tcm.com/tcmdb/title/2025207/Go-Go-Boys-The-Inside-Story-of-Cannon-Films-The/|title=OVERVIEW|access-date=October 20, 2014}}&lt;/ref&gt;
*[[Sylvester Stallone]] as Party Guest (uncredited){{efn|The then-unknown [[Sylvester Stallone]] was an extra in this movie during the party scenes.&lt;ref&gt;{{cite web |title=The Sidelong Glances of a Pigeon Kicker (1970) |url=https://www.imdb.com/title/tt0067576/mediaviewer/rm1105675520 |website=[[IMDb]] |accessdate=22 August 2018}}&lt;/ref&gt; He may be seen in several shots wearing the same colorful sweater he wore in an adult film he also made in 1970, ''[[The Party at Kitty and Stud's]]'', also known as ''Italian Stallion.''&lt;ref&gt;http://img21.mtime.cn/mg/2011/04/05/023047.11375385.jpg {{Bare URL image|date=March 2022}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.neatorama.com/2015/05/06/Sylvester-Stallones-First-Film-The-Party-at-Kitty-and-Studs/|title=Sylvester Stallone's First Film: The Party at Kitty and Stud's|website=Neatorama|date=6 May 2015 |accessdate=May 13, 2019}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=http://www.stallonezone.com/main/1970/10/sly-scores.html|title=StalloneZone: Sly Scores|accessdate=May 13, 2019}}&lt;/ref&gt;}}
* [[Sylvester Stallone]] - Stallone had his first starring role in the [[softcore pornography]] feature film ''[[The Party at Kitty and Stud's]]'' (1970). He was paid $200 for two days' work. The production company later changed the title of the movie to [[The Party at Kitty and Stud's|The Italian Stallion]] in order to capitalize on Stallone's success in his break out role in ''[[Rocky]]''.&lt;ref name=AVNhoax1&gt;{{cite web|last=Sullivan|first=David|date=6 February 2008|title=The 'Italian Stallion' hoax: Stallone never did hardcore|website=AVN Business|publisher=[[Adult Video News]]|url=http://business.avn.com/articles/28745.html|accessdate=9 November 2017|archivedate=23 August 2009|archiveurl=https://web.archive.org/web/20090823060517/http://business.avn.com/articles/28745.html}}&lt;/ref&gt;
| children          = 5, including [[Sylvester Stallone|Sylvester]] and [[Frank Stallone Jr.|Frank&amp;nbsp;Jr.]]
'''Francesco Stallone''' (September 12, 1919 &amp;ndash; July 11, 2011)&lt;ref&gt;[https://books.google.com/books?id=SAqdwE8hZKsC&amp;dq=frank+stallone+1919+2011&amp;pg=PA328 ''Obituaries in the Performing Arts'']&lt;/ref&gt; was an Italian-American hairdresser.&lt;ref&gt;{{cite web|url=http://www.legacy.com/ns/obituary.aspx?n=frank-stallone&amp;pid=152530892|title=Frank Stallone Sr., 91, polo enthusiast and father of actor Sylvester Stallone|publisher=www.legacy.com|accessdate=October 29, 2014}}&lt;/ref&gt; His children include actors [[Sylvester Stallone]] and [[Frank Stallone Jr.]]&lt;ref&gt;{{cite web|url=http://www.upi.com/Entertainment_News/Movies/2011/07/14/Frank-Stallone-Sr-dead-at-91/UPI-94281310671596/|title=Frank Stallone Sr. dead at 91|publisher=www.upi.com|accessdate=October 29, 2014}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.sylvesterstallone.com/frank-stallone-sr-1919-2011/|title=Frank Stallone Sr. (1919-2011)|publisher=www.sylvesterstallone.com|accessdate=October 29, 2014}}&lt;/ref&gt; Stallone Sr. wrote ''Stewart Lane'' which was published in May 2010.&lt;ref&gt;{{cite web|url=https://www.amazon.com/Frank-Stallone-Sr./e/B00JNFB5EI|title=Books by Frank Stallone Sr.|publisher=Amazon|accessdate=October 29, 2014}}&lt;/ref&gt; He appeared in the 1976 film ''[[Rocky]]'' as the timekeeper.
Stallone was born in [[Gioia del Colle]], to Silvestro Stallone (1883-1963), a barber, and Pulcheria Nicastri (1890-1973).&lt;ref&gt;[https://www.antenati.san.beniculturali.it/ark:/12657/an_ua4145714/wb7yoXZ Archivio di Stato di Bari, civil status of Gioia del Colle, birth certificate No. 734 of November 10, 1883.]&lt;/ref&gt;&lt;ref&gt;[https://www.antenati.san.beniculturali.it/ark:/12657/an_ua4152345/LPn9rA1 Archivio di Stato di Bari, civil status of Gioia del Colle, birth certificat No. 167 of February 21, 1890]&lt;/ref&gt;&lt;ref&gt;Archivio di Stato di Bari, civil status of Gioia del Colle, marriage certificate No. 134 of November 15, 1909.&lt;/ref&gt; He was one of eight children. His family moved to New York from Italy in 1923 on board the [[SS Dante Alighieri]] sailing from Naples.&lt;ref&gt;New York Passenger Arrival Lists (Ellis Island), 1892-1924 Roll 3352, vol 7667-7668, 17 Aug 1923-18 Aug 1923&lt;/ref&gt; Stallone served in the [[United States Army|US Army]] during [[World War II]] from 1940-1945.&lt;ref&gt;{{cite web|url=http://mobile.philly.com/blogs/?wss=/philly/blogs/phillygossip/&amp;id=125478538|title=Frank Stallone Sr., father of Frank and Sylvester Stallone, dies at 91|publisher=mobile.philly.com|accessdate=October 29, 2014|archive-date=November 12, 2014|archive-url=https://web.archive.org/web/20141112132533/http://mobile.philly.com/blogs/?wss=/philly/blogs/phillygossip/&amp;id=125478538|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://lawprofessors.typepad.com/immigration/2011/07/immigrant-of-the-day-frank-stallone-sr-italy.html|title=Immigrant of the Day: Frank Stallone Sr. (Italy)|accessdate=October 29, 2014}}&lt;/ref&gt;
Stallone had a short-lived career as an actor and as a writer. His only acting role was in the role of the timekeeper in the 1976 American sports drama ''[[Rocky]]''.&lt;ref&gt;{{cite web|url=http://www.movieneon.com/titles/1976/rocky/characters/timekeeper/|title=&quot;Timekeeper&quot; (Frank Stallone Sr.) in Rocky (1976)|publisher=www.movieneon.com|accessdate=October 30, 2014}}&lt;/ref&gt; which starred his son [[Sylvester Stallone|Sylvester]].&lt;ref&gt;{{cite web|url=http://www.filmsite.org/rock.html|title=Filmsite Movie Review &quot;Rocky&quot; (1976)|publisher=www.filmsite.org|accessdate=October 30, 2014}}&lt;/ref&gt; At the age of 90, he wrote his first and only novel, which was published on May 24, 2010.&lt;ref&gt;{{cite book|title=Stewart Lane by Frank Stallone, Sr|isbn=978-1450211369|last1=Frank Stallone|first1=Sr|date=May 2010|publisher=iUniverse }}&lt;/ref&gt; The book, entitled ''Stewart Lane'', was the tale of a couple who attempt to renovate a dilapidated country house.&lt;ref&gt;{{cite web|url=http://www.digitalspy.com/celebrity/news/a330040/sylvester-stallones-father-frank-sr-dies-aged-91.html#~oU6Wdrh6xd05kX|title=Sylvester Stallone's father Frank Sr dies, aged 91|date=15 July 2011|publisher=www.digitalspy.com|accessdate=October 30, 2014}}&lt;/ref&gt;
Stallone had five children: [[Sylvester Stallone]] and [[Frank Stallone|Frank Jr.]] (with Jackie); Bryan and Carla Francesca (with Rose Marie);&lt;ref&gt;{{cite web|url=http://dalje.com/en-bestseller/frank-stallone-sr-dead-at-91/370939|title=Frank Stallone Sr. dead at 91|publisher=dalje.com|accessdate=October 30, 2014|url-status=dead|archiveurl=https://web.archive.org/web/20141030125531/http://dalje.com/en-bestseller/frank-stallone-sr-dead-at-91/370939|archivedate=October 30, 2014}}&lt;/ref&gt;&lt;ref name=&quot;Trivia&quot;&gt;{{cite web|url=https://m.imdb.com/name/nm2200613/trivia?ref_=m_nm_dyk_trv|title=Frank Stallone: Trivia|publisher=www.imdb.com|accessdate=October 30, 2014}}&lt;/ref&gt; and Dante Alexander (with Kathleen). Among his grandchildren are [[Sage Stallone|Sage]] and [[Sistine Stallone]].
[[Category:Family of Sylvester Stallone]]</text>
```



2708 In May 2019, Amirpour announced that she would have directed a female-led reboot of the 1993 film ''[[Cliffhanger (film)|Cliffhanger]]''. [[Jason Momoa]] was a principal actor.&lt;ref&gt;{{Cite web|url = https://deadline.com/2019/05/cliffhanger-reboot-jason-momoa-ana-lily-amirpour-cannes-female-1202608726/|title = Hang on, Cannes! Neal Moritz's Female-Fronted 'Cliffhanger' Reboot Climbs with Rocket Science, CAA|date = May 8, 2019|access-date = May 29, 2019|archive-date = May 8, 2019|archive-url = https://web.archive.org/web/20190508222223/https://deadline.com/2019/05/cliffhanger-reboot-jason-momoa-ana-lily-amirpour-cannes-female-1202608726/|url-status = live}}&lt;/ref&gt; But by May 2023, it was officially reported that the reboot will be redeveloped as a legacy-sequel instead. [[Ric Roman Waugh]] replaced Amirpour as director and [[Sylvester Stallone]] returned to reprise his role from the original, in addition to taking on a role as producer. The sequel will have a presence at Cannes, where distributing studios will be decided. No release date has been set.&lt;ref name=&quot;Sequel_Deadline&quot;&gt;{{cite web|url=https://deadline.com/2023/05/sylvester-stallone-set-for-cliffhanger-reboot-from-director-ric-roman-waugh-1235351858/|work=Deadline|title=Sylvester Stallone Set For 'Cliffhanger' Reboot From Director Ric Roman Waugh|author=Grobar, Matt|date=May 1, 2023|accessdate=June 1, 2023|archive-date=June 1, 2023|archive-url=https://web.archive.org/web/20230601095319/https://deadline.com/2023/05/sylvester-stallone-set-for-cliffhanger-reboot-from-director-ric-roman-waugh-1235351858/|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Sequel_THR&quot;&gt;{{cite web|url=https://www.hollywoodreporter.com/movies/movie-news/cliffhanger-reboot-set-sylvester-stallone-1235476408/|work=The Hollywood Reporter|title=Sylvester Stallone Sets 'Cliffhange'' Reboot With 'Angel Has Fallen' Director|author=Kit, Borys|date=May 1, 2023|accessdate=June 1, 2023|archive-date=May 31, 2023|archive-url=https://web.archive.org/web/20230531003844/https://www.hollywoodreporter.com/movies/movie-news/cliffhanger-reboot-set-sylvester-stallone-1235476408/|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;Sequel_Variety&quot;&gt;{{cite web|url=https://variety.com/2023/film/news/sylvester-stallone-cliffhanger-reboot-ric-roman-waugh-1235599620/|work=Variety|title=Sylvester Stallone Returning for 'Cliffhanger' Reboot, Ric Roman Waugh Directing|author=Lang, Brent|date=May 1, 2023|accessdate=June 1, 2023|archive-date=June 1, 2023|archive-url=https://web.archive.org/web/20230601095320/https://variety.com/2023/film/news/sylvester-stallone-cliffhanger-reboot-ric-roman-waugh-1235599620/|url-status=live}}&lt;/ref&gt;

2709 Dickinson remembers relationships with fellow celebrities including [[Sylvester Stallone]], [[Liam Neeson]], [[Warren Beatty]], [[Mick Jagger]], and [[Jack Nicholson]].&lt;ref name=&quot;publishersweekly&quot; /&gt; She enjoys her celebrity status and her time with friends [[John Belushi]], [[Truman Capote]], and [[Andy Warhol]].&lt;ref name=&quot;wallace&quot; /&gt; Belushi and Dickinson enable each other's predilections for abusing drugs.&lt;ref name=&quot;weiss&quot; /&gt; She regularly spends time at [[Studio 54]] during this period.&lt;ref name=&quot;wallace&quot; /&gt; Dickinson recounts a darker side of celebrity, and writes that she trusted in [[Bill Cosby]] when he informed her he could help her career in [[show business]], only to experience verbal anger from him when she said she was too tired to engage in sexual activity with him.&lt;ref name=&quot;weiss&quot; /&gt;

2710 She is in the music video for the song &quot;[[Genghis Khan (Miike Snow song)|Genghis Khan]]&quot; by [[Miike Snow]], in the role of the wife of the Bond-villain character. She has worked on advertising campaigns for [[Warburtons]] as recurring character Michelle co-starring [[Robert de Niro]], [[Sylvester Stallone]], [[The Muppets]] and [[Peter Kay]],&lt;ref&gt;{{cite news|url=https://www.burnleyexpress.net/whats-on/theatre-and-comedy/caroline-o-hara-the-coronation-street-star-on-her-acting-journey-and-championing-burnley-empire-theatre-1-9578150 |title=Caroline O'Hara: the Coronation Street star on her acting journey and championing Burnley Empire Theatre|newspaper=Burnley Express |date=2019-02-06 |accessdate=2019-03-14}}&lt;/ref&gt; also for [[House of Fraser]],&lt;ref&gt;{{cite web|url=https://m.youtube.com/watch?v=UEhAwokcqN8 |title=The Fairy Tale &amp;#124; It all starts with the dress |publisher=YouTube |date=2014-11-21 |accessdate=2015-12-26}}&lt;/ref&gt; playing Cinderella, and was cast in a series of adverts for [[Walker's Crisps]] 'to get more young men to buy its crisps'.&lt;ref&gt;{{cite web|url=http://www.campaignlive.co.uk/thework/1288598/ |title=Walkers &quot;go big or go home&quot; by Abbott Mead Vickers BBDO |publisher=Campaignlive.co.uk |date=2014-04-02 |accessdate=2015-12-26}}&lt;/ref&gt;

2711 * [[Sylvester Stallone]]

2712 '''''Animal Crackers''''' is a 2017 animated [[comedy film|comedy]]-[[fantasy film]] directed by [[Scott Christian Sava]] and [[Tony Bancroft]], written by Sava and [[Dean Lorey]] and based on the [[animal cracker|animal-shaped cookie]] (and also loosely on the graphic novel by Sava).&lt;ref&gt;{{Cite web|url = https://bluedreamstudios.com/2017/07/22/animal-crackers-first-official-trailer/|title = Animal Crackers|date = July 22, 2017|access-date = July 25, 2020|archive-date = September 28, 2020|archive-url = https://web.archive.org/web/20200928211923/https://bluedreamstudios.com/2017/07/22/animal-crackers-first-official-trailer/|url-status = dead}}&lt;/ref&gt; The film features the voices of [[Emily Blunt]], [[John Krasinski]], [[Danny DeVito]], [[Ian McKellen]], [[Sylvester Stallone]], [[Patrick Warburton]], [[Raven-Symoné]], [[Harvey Fierstein]], [[Wallace Shawn]], [[Gilbert Gottfried]], [[Tara Strong]], [[James Arnold Taylor]], [[Kevin Grevioux]], and Lydia Rose Taylor in her film debut. It tells the story of a family who comes across a box of magical animal crackers that turns anyone that consumes a cracker into the animal that the cracker represents and this animal handily saves the circus that the family was associated with.

2713 * [[Sylvester Stallone]] as Bulletman, the [[human cannonball]] in a bullet-shaped helmet who only says his name until the end of the movie.&lt;ref name=thewrap /&gt;

2714 On November 6, 2014, Blue Dream Studios announced [[Sylvester Stallone]], [[Danny DeVito]], and [[Ian McKellen]] as lead voice cast.&lt;ref name=thewrap&gt;{{cite web|url=https://www.thewrap.com/sylvester-stallone-danny-devito-ian-mckellen-lead-animal-crackers-voice-cast/|title=Sylvester Stallone, Danny DeVito, Ian McKellen Lead 'Animal Crackers' Voice Cast|publisher=[[TheWrap]]|author=Jeff Sneider|date=November 6, 2014|access-date=December 29, 2014}}&lt;/ref&gt; On February 3, 2015, [[John Krasinski]] and [[Kaley Cuoco]] joined the cast as Owen and Zoe Huntington, respectively.&lt;ref name=johnkaleycast&gt;{{cite web|url=https://deadline.com/2015/02/animal-crackers-adds-kaley-cuoco-john-krasinski-to-voice-cast-berlin-1201365521/|title='Animal Crackers' Adds Kaley Cuoco &amp; John Krasinski To Voice Cast - Berlin|website=[[Deadline Hollywood]]|author=Dominio Patten|date=February 3, 2015|access-date=February 4, 2015}}&lt;/ref&gt; On March 30, 2015, [[Emily Blunt]] replaced Cuoco due to a scheduling conflict.&lt;ref name=DeadlineEmilyBlunt&gt;{{cite web|last1=D'Alessandro|first1=Anthony|title=Emily Blunt Takes Bite Of 'Animal Crackers'; 'Neon Demon' Cast Is Set - Film Briefs|url=https://deadline.com/2015/03/emily-blunt-animal-crackers-neon-demon-nicolas-winding-refn-1201401223/|website=Deadline Hollywood|access-date=August 3, 2015|date=March 30, 2015}}&lt;/ref&gt;

2715 |ShortSummary=Sylvester Stallone Phone Call; Kamal Gray advertises social faux pas; Jimmy advertises writers' book; ''Tonight Show'' Pros &amp; Cons: Watching ''[[Empire (2015 TV series)|Empire]]''; ''Tonight Show'' Kids Letters; Jimmy reads Oscar cue cards; Gza with Tom Morello performed &quot;The Mexican&quot;

2716 {{User:COIBot/EditSummary|id=42847176|lang=es|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Gemsinrock|link=www.celebheightweight.com/2015/01/sylvester-stallone-height-weight-age-measurements-net-worth/|sortdomain=com.celebheightweight.|domain=celebheightweight.com|origdiff=http://es.wikipedia.org/w/index.php?diff=79224276&amp;oldid=79012917|resolved=X|isIP=0|date=2015-01-05|time=12:09:04|wiki=es.wikipedia.org|revid=79224276|oldid=79012917|usercount=4|whitelisted=0|blacklisteduser=0|whitereason=|blackreason=|deleted=0|top=0|there=|checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=www.celebheightweight.com/2015/01/sylvester-stallone-height-weight-age-measurements-net-worth/|domain=celebheightweight.com|U=4|L=209|UL=2|WUL=0|base=|basedomain=|baseip=}}}}

2717 He was named CEO of [[Miramax]] in April 2017.&lt;ref name=&quot;variety&quot;/&gt; He left the studio in October 2023 following the lapse of his contract with them, forming another production company, BlockFilm, later that month.&lt;ref&gt;{{cite web |url= https://deadline.com/2023/10/miramax-ceo-bill-block-exits-1235561838/|title= Shocker! Shakeup At Miramax As CEO Bill Block Exits|date= October 2, 2023|access-date= October 2, 2023|first1= Mike Jr.|last1= Fleming|first2= Nellie|last2= Andreeva}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=Jason Statham To Star In David Ayer-Directed Sylvester Stallone Script 'Levon's Trade' From Black Bear &amp; Bill Block's BlockFilm – AFM Hot Package|website=[[Deadline Hollywood]]|first=Andreas|last=Wiseman|date=27 October 2023|access-date=28 October 2023|url=https://deadline.com/2023/10/jason-statham-david-ayer-sylvester-stallone-levons-trade-black-bear-bill-block-afm-1235585741/}}&lt;/ref&gt; Producer [[Jonathan Glickman]] subsequently replaced Block as Miramax's CEO on April 2, 2024.&lt;ref&gt;{{cite web|title=Miramax Names Jonathan Glickman CEO|website=[[Deadline Hollywood]]|first=Anthony|last=D'Alessandro|date=2 April 2024|access-date=2 April 2024|url=https://deadline.com/2024/04/miramax-jonathan-glickman-1235874073/}}&lt;/ref&gt;

2718 | alt   = At a boxing ring in a gym, Rocky Balboa (Sylvester Stallone) looks at Donnie Creed (Michael B. Jordan) with Rocky's left hand resting on Creed's right shoulder. The films tagline reads &quot;Your Legacy is more than a name&quot; with credits and the film's title on the top.

2719 | based_on   = {{Based on|[[List of Rocky characters|Characters]]|[[Sylvester Stallone]]}}

2720 * Sylvester Stallone

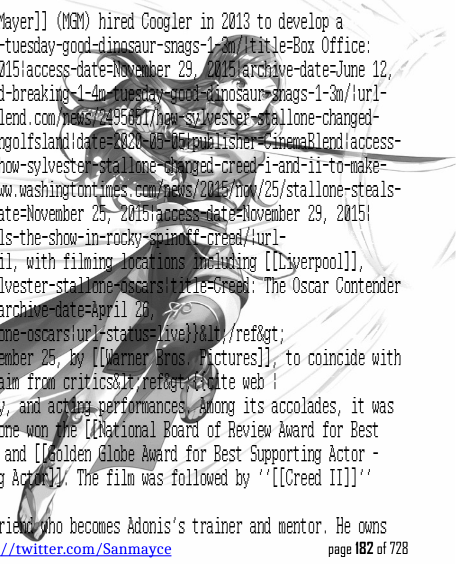
2721 * Sylvester Stallone

2722 '''''Creed''''' is a 2015 American [[Sports film|sports]] [[Drama (film and television)|drama film]] directed by [[Ryan Coogler]], who co-wrote the screenplay with [[Aaron Covington]]. It is the first spin-off of and is the seventh installment in the [[Rocky (franchise)|''Rocky'' film series]]. It stars [[Michael B. Jordan]], [[Sylvester Stallone]], [[Tessa Thompson]], [[Phylicia Rashad]], [[Tony Bellew]], and [[Graham McTavish]]. In the film, amateur boxer [[Adonis Creed]] (Jordan) is trained and mentored by [[Rocky Balboa]] (Stallone), the former rival turned friend of Adonis' father, [[Apollo Creed]].

2723 Although ''[[Rocky Balboa (film)|Rocky Balboa]]'' (2006) was considered the end of the franchise, [[Metro-Goldwyn-Mayer Pictures|Metro-Goldwyn-Mayer]] (MGM) hired Coogler in 2013 to develop a ''Rocky'' spin-off.&lt;ref&gt;{{cite web|url=https://www.forbes.com/sites/scottmendelson/2015/11/25/box-office-creed-rocks-record-breaking-1-4m-tuesday-good-dinosaur-snags-1-3m/|title=Box Office: 'Creed' Rocks Record-Breaking $1.4M Tuesday, Pixar's 'The Good Dinosaur' Snags $1.3M|author=Scott Mendelson|work=[[Forbes]]|date=November 25, 2015|access-date=November 29, 2015|archive-date=June 12, 2018|archive-url=https://web.archive.org/web/20180612164335/https://www.forbes.com/sites/scottmendelson/2015/11/25/box-office-creed-rocks-record-breaking-1-4m-tuesday-good-dinosaur-snags-1-3m/|url-status=live}}&lt;/ref&gt; Coogler and Covington's original screenplay was completed by April 2014,&lt;ref&gt;{{cite web|url=https://www.cinemablend.com/news/2495651/how-sylvester-stallone-changed-creed-i-and-ii-to-make-it-more-comfortable|title=How Sylvester Stallone Changed Creed I And II To Make It More 'Comfortable'|first=Jason|last=Ingolfsland|date=2020-05-05|publisher=CinemaBlend|access-date=2020-11-18|archive-date=November 30, 2020|archive-url=https://web.archive.org/web/20201130110607/https://www.cinemablend.com/news/2495651/how-sylvester-stallone-changed-creed-i-and-ii-to-make-it-more-comfortable|url-status=live}}&lt;/ref&gt; with the involvement of Jordan and Stallone also confirmed.&lt;ref&gt;{{cite web|url=http://www.washingtontimes.com/news/2015/nov/25/stallone-steals-the-show-in-rocky-spinoff-creed/|title=REVIEW: 'Creed': half-Rocky sequel, half-Rocky spinoff|author=Josh Terry|work=[[The Washington Times]]|date=November 25, 2015|access-date=November 29, 2015|archive-date=July 6, 2018|archive-url=https://web.archive.org/web/20180706075328/https://www.washingtontimes.com/news/2015/nov/25/stallone-steals-the-show-in-rocky-spinoff-creed/|url-status=live}}&lt;/ref&gt; The remaining cast were hired by January 2015, after which, [[principal photography]] began and lasted until that April, with filming locations including [[Liverpool]], [[Philadelphia]], and [[Gainesville, Georgia]].&lt;ref&gt;{{cite web|url=https://www.vanityfair.com/hollywood/2015/11/creed-michael-b-jordan-sylvester-stallone-oscars|title=Creed: The Oscar Contender We Should Have Seen Comingf|author=Katey Rich|work=[[Vanity Fair (magazine)|Vanity Fair]]|date=November 18, 2015|access-date=November 29, 2015|archive-date=April 26, 2018|archive-url=https://web.archive.org/web/20180426024344/https://www.vanityfair.com/hollywood/2015/11/creed-michael-b-jordan-sylvester-stallone-oscars|url-status=live}}&lt;/ref&gt;

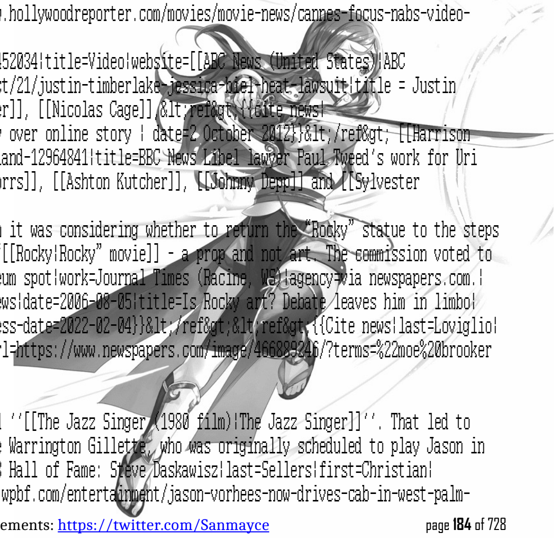
2724 ''Creed'' had its premiere on November 19, 2015, at the [[Regency Village Theater]] in Los Angeles and was released in the United States on November 25, by [[Warner Bros. Pictures]], to coincide with the 40th anniversary of the date of the opening scene in the [[Rocky|first film]]. It grossed $173&amp;nbsp;million worldwide and received acclaim from critics&lt;ref&gt;{{cite web | url=https://www.metacritic.com/movie/creed | title=Creed | website=[[Metacritic]] }}&lt;/ref&gt; who praised Coogler's direction, the screenplay, and acting performances. Among its accolades, it was selected by [[National Board of Review]] as one of the [[National Board of Review Awards 2015#Top 10 Films|top ten films of 2015]], while Stallone won the [[National Board of Review Award for Best Supporting Actor]], the [[Broadcast Film Critics Association Award for Best Supporting Actor|Critics' Choice Award for Best Supporting Actor]], and [[Golden Globe Award for Best Supporting Actor – Motion Picture|Golden Globe Award for Best Supporting Actor in a Motion Picture]], and was nominated for the [[Academy Award for Best Supporting Actor]]. The film was followed by ''[[Creed II]]'' (2018) and ''[[Creed III]]'' (2023). The film is the first film in the franchise to have been produced by [[Sylvester Stallone]].

2725 * [[Sylvester Stallone]] as [[Rocky Balboa|Robert &quot;Rocky&quot; Balboa]]: A two-time world heavyweight champion and Apollo's rival-turned-friend who becomes Adonis's trainer and mentor. He owns



```
and operates an Italian restaurant in Philadelphia named after his deceased wife [[Adrian Pennino|Adrian]] (played by [[Talia Shire]] in previous films).&lt;ref name=&quot;CooglerTHR&quot; /&gt;
[[File:Sylvester Stallone, Tessa Thompson, and Michael B. Jordan promoting Creed at the Philadelphia Art Museum.JPG|thumb|right|Stallone, Thompson, and Jordan promoting the film atop the [[Rocky Steps]] in November 2015.]]
On July 24, 2013, it was announced that [[Metro-Goldwyn-Mayer]] (MGM) had signed on with ''[[Fruitvale Station]]'' (2013) director [[Ryan Coogler]] to direct a spin-off of ''[[Rocky]]'' (1976), a seventh film in the [[Rocky (franchise)|''Rocky'' series]], which Coogler would also co-write with [[Aaron Covington]].&lt;ref name=DeadlineCoogler&gt;{{cite news|last1=Fleming|first1=Mike Jr.|title='Fruitvale Station' Duo Ryan Coogler And Michael B. Jordan Team With Sly Stallone On MGM 'Rocky' Spinoff 'Creed'|url=https://deadline.com/2013/07/fruitvale-station-duo-ryan-coogler-and-michael-b-jordon-team-with-sly-stallone-on-mgm-rocky-spinoff-creed-547901/|access-date=February 2, 2015|website=Deadline Hollywood|date=July 24, 2013|archive-date=February 14, 2015|archive-url=https://web.archive.org/web/20150214055724/http://deadline.com/2013/07/fruitvale-station-duo-ryan-coogler-and-michael-b-jordon-team-with-sly-stallone-on-mgm-rocky-spinoff-creed-547901/|url-status=live}}&lt;/ref&gt; [[Sylvester Stallone]] also worked on the screenplay for the seventh film.&lt;ref name=&quot;writers1&quot;&gt;{{cite web|url=https://www.youtube.com/watch?v=k4-tUD_MkJk|title=CREED Screenwriter Aaron Covington on developing the script &amp; Sylvester Stallone's legacy as ROCKY|work=Jog Road Productions|date=December 7, 2015|access-date=December 15, 2015|archive-date=March 10, 2016|archive-url=https://web.archive.org/web/20160310102140/https://www.youtube.com/watch?v=k4-tUD_MkJk|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;writers2&quot;&gt;{{cite news|url=http://www.torontosun.com/2014/12/31/sylvester-stallone-potentially-leaks-creed-spoiler-online|title=Sylvester Stallone potentially leaks 'Creed' spoiler online|quote=The Expendables star has been working on the screenplay for new film Creed|newspaper=[[Toronto Sun]]|date=December 31, 2014|access-date=November 29, 2015|archive-date=December 8, 2015|archive-url=https://web.archive.org/web/20151208165951/http://www.torontosun.com/2014/12/31/sylvester-stallone-potentially-leaks-creed-spoiler-online|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;writers3&quot;&gt;{{cite web|url=https://www.yahoo.com/movies/sylvester-stallone-may-have-just-spoiled-the-106515717367.html|title=Sylvester Stallone May Have Just Spoiled The Ending of the New 'Rocky' Movie|quote=Stallone's apparently making edits to the script, since the tweet's accompanying text read, 'Where the screenwriting is done.'|first=Jordan|last=Zakarin|work=[[Yahoo! Movies]]|date=December 29, 2014|access-date=November 29, 2015|archive-date=December 8, 2015|archive-url=https://web.archive.org/web/20151208055833/https://www.yahoo.com/movies/sylvester-stallone-may-have-just-spoiled-the-106515717367.html|url-status=live}}&lt;/ref&gt; The film would focus on a man following in the footsteps of his late father, [[Apollo Creed]], and getting a mentor in the now-retired [[Rocky Balboa]]. [[Michael B. Jordan]] was set for the role of Creed's son, Adonis &quot;Donnie&quot; Creed,&lt;ref&gt;{{cite web | title=BREAKING NEWS re: Creed Movie - Michael B. Jordan and Sylvester Stallones - SPOILER ALERT | work=phillychitchat.com | url=http://www.phillychitchat.com/2015/02/breaking-news-re-creed-movie-michael-b.html | access-date=February 24, 2015 | archive-url=https://web.archive.org/web/20150924072352/http://www.phillychitchat.com/2015/02/breaking-news-re-creed-movie-michael-b.html | archive-date=September 24, 2015 | url-status=dead }}&lt;/ref&gt; and Stallone was set to reprise his character of Rocky.&lt;ref name=&quot;DeadlineCoogler&quot; /&gt; Original producers [[Irwin Winkler]] and [[Robert Chartoff]] would produce, along with Stallone and Kevin King-Templeton.&lt;ref name=&quot;DeadlineCoogler&quot; /&gt; On April 25, 2014, while talking to ''[[The Hollywood Reporter]]'', Coogler stated that he had sent his latest draft to the studio, and confirmed the involvement of Jordan and Stallone.&lt;ref name=CooglerTHR&gt;{{cite news|last1=Kemp|first1=Stuart|title=Sundance London: Ryan Coogler Talks 'Rocky' Spinoff 'Creed'|url=https://www.hollywoodreporter.com/heat-vision/sundance-london-ryan-coogler-talks-698869|access-date=February 2, 2015|work=The Hollywood Reporter|date=April 25, 2014|archive-date=February 14, 2015|archive-url=https://web.archive.org/web/20150214051440/http://www.hollywoodreporter.com/heat-vision/sundance-london-ryan-coogler-talks-698869|url-status=live}}&lt;/ref&gt;
In an interview with Ellen, Sylvester Stallone reasoned why he refused to write ''Creed''. He referred to a generational gap between the time when he wrote ''Rocky'' and the time when a film such as ''Creed'' would be making its appearance on the silver screen.&lt;ref&gt;{{cite web |last1=Sardar |first1=Samrat |title=Sylvester Stallone Refused to Write 'Creed' Because of 4-Decade-Long Problem That Could Not Be Fixed |url=https://www.essentiallysports.com/boxing-news-sylvester-stallone-refused-to-write-creed-because-of-4-decade-long-problem-that-could-not-be-fixed/ |website=EssentiallySports |date=December 29, 2022 |access-date=December 29, 2022}}&lt;/ref&gt;
[[Principal photography]] began on January 19, 2015 on location at Goodison Park, with the first scene shot taking place during a [[Barclays Premier League]] football match between [[Everton F.C.|Everton]] (of which Stallone and native Evertonian Bellew are fans) and [[West Bromwich Albion F.C.|West Bromwich Albion]].&lt;ref&gt;{{cite news|last1=Owen|first1=Luke|title=Rocky spin-off Creed to shoot tonight at Goodison Park during Everton v West Brom game|url=http://www.flickeringmyth.com/2015/01/rocky-spin-off-creed-shoot-tonight-goodison-park-everton-v-west-brom-game.html|access-date=January 20, 2015|publisher=flickeringmyth.com|date=January 19, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite news|title=Sylvester Stallone: Rocky star delivers message at Everton|url=https://www.bbc.com/sport/0/football/30891678|access-date=January 20, 2015|publisher=bbc.com|date=January 19, 2015|archive-date=January 21, 2015|archive-url=https://web.archive.org/web/20150121074824/http://www.bbc.com/sport/0/football/30891678|url-status=live}}&lt;/ref&gt; Goodison would later host both the climactic film fight between Adonis and Conlan and also Bellew's real-life title fight against Ilunga Makabu in May 2016, which was the first outdoor boxing match in Liverpool since 1949.
On February 13, the crew was spotted filming in the Victor Cafe in [[South Philadelphia]].&lt;ref&gt;{{cite news|title=The new 'Rocky' movie 'Creed' filming at The Victor Cafe in Philly|url=http://www.onlocationvacations.com/2015/02/13/the-new-rocky-movie-creed-filming-at-the-victor-cafe-in-philly/|access-date=February 14, 2015|publisher=onlocationvacations.com|date=February 13, 2015|archive-url=https://web.archive.org/web/20150702111431/http://www.onlocationvacations.com/2015/02/13/the-new-rocky-movie-creed-filming-at-the-victor-cafe-in-philly/|archive-date=July 2, 2015|url-status=dead}}&lt;/ref&gt; The café was transformed into the &quot;Adrian's Restaurant&quot;, and crew were again spotted filming there on February 16.&lt;ref&gt;{{cite news|title=See Philadelphia's Victor Cafe transformed into Adrien's for 'Creed'|url=http://www.onlocationvacations.com/2015/02/16/see-philadelphias-victor-cafe-transformed-into-adriens-for-creed/|access-date=February 17, 2015|publisher=onlocationvacations.com|date=February 16, 2015|archive-date=February 17, 2015|archive-url=https://web.archive.org/web/20150217065707/http://www.onlocationvacations.com/2015/02/16/see-philadelphias-victor-cafe-transformed-into-adriens-for-creed/|url-status=live}}&lt;/ref&gt; Stallone and Jordan were also spotted on the set of the film on February 18.&lt;ref&gt;{{cite news|title=Sylvester Stallone and Michael B. Jordan on the Creed Set|url=https://www.comingsoon.net/movies/news/411207-sylvester-stallone-and-michael-b-jordan-on-the-creed-set#/slide/9|access-date=February 19, 2015|publisher=comingsoon.net|date=February 18, 2015|archive-date=February 19, 2015|archive-url=https://web.archive.org/web/20150219135103/http://www.comingsoon.net/movies/news/411207-sylvester-stallone-and-michael-b-jordan-on-the-creed-set#/slide/9|url-status=live}}&lt;/ref&gt; From February 24-27 and then on March 3, filming took place at [[Sun Center Studios]] in [[Aston Township, Pennsylvania|Aston Township]].&lt;ref&gt;{{cite news|title=Filming for the new 'Rocky' movie 'Creed' moves to Aston, PA this week|url=http://www.onlocationvacations.com/2015/02/24/filming-for-the-new-rocky-movie-creed-moves-to-aston-pa-this-week/|access-date=February 25, 2015|publisher=onlocationvacations.com|date=February 24, 2015|archive-date=February 25, 2015|archive-url=https://web.archive.org/web/20150225181641/http://www.onlocationvacations.com/2015/02/24/filming-for-the-new-rocky-movie-creed-moves-to-aston-pa-this-week/|url-status=live}}&lt;/ref&gt;
The screenplay was ranked the 22nd best American screenplay of the 21st century in [[IndieWire]], with Zack Sharf writing, &quot;Not only does the script manage to tell an authentic origin story of the young and determined Adonis Creed, but it also finds an authentic way to revive Sylvester Stallone's Rocky and make him the emotional lynchpin of Adonis' rise to champion boxer.&quot;&lt;ref&gt;{{Cite web|last1=Dry|first1=Jude|last2=O'Falt|first2=Chris|last3=Erbland|first3=Kate|last4=Kohn|first4=Eric|last5=Sharf|first5=Zack|last6=Marotta|first6=Jenna|last7=Thompson|first7=Anne|last8=Earl|first8=William|last9=Nordine|first9=Michael|last10=Ehrlich|first10=David|date=April 20, 2018|title=The 25 Best American Screenplays of the 21st Century, From 'Eternal Sunshine' to 'Lady Bird'|url=https://www.indiewire.com/2018/04/best-screenplays-scripts-american-1201952200/|access-date=February 4, 2021|website=IndieWire|language=en}}&lt;/ref&gt;
| [[Sylvester Stallone]]
| [[Sylvester Stallone]]
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
| Sylvester Stallone
In January 2016, Sylvester Stallone and MGM CEO [[Gary Barber]] confirmed that a sequel to ''Creed'' was in development.&lt;ref&gt;{{cite news|last1=Setoodeh|first1=Ramin|title=Sylvester Stallone Says 'Creed' Sequel Could Reunite Rocky and Apollo (EXCLUSIVE)|url=https://variety.com/2016/film/news/creed-sequel-sylvester-stallone-1201672298/|access-date=January 5, 2016|publisher=Variety|date=January 5, 2016|archive-date=January 6, 2016|archive-url=https://web.archive.org/web/20160106052846/http://variety.com/2016/film/news/creed-sequel-sylvester-stallone-1201672298/|url-status=live}}&lt;/ref&gt; That same month, Stallone posed the possibility of seeing [[Milo Ventimiglia]] appear in the sequel, reprising his role as Rocky's son Robert Balboa from ''[[Rocky Balboa (film)|Rocky Balboa]]''. Ventimiglia previously revealed during the development of ''Creed'' that he was open to returning to the franchise, stating, &quot;I'll tell you what, if they invited me, I'd love to be there. If they didn't, I wouldn't be offended.&quot;&lt;ref&gt;{{cite news|last1=Topel|first1=Fred|title=Exclusive: Milo Ventimiglia Ready for Creed Comeback|url=http://www.craveonline.com/site/545137-exclusive-milo-ventimiglia-ready-for-creed-comeback|access-date=September 28, 2017|publisher=Crave|date=August 3, 2013|archive-url=https://web.archive.org/web/20170928152823/http://www.craveonline.com/site/545137-exclusive-milo-ventimiglia-ready-for-creed-comeback|archive-date=September 28, 2017|url-status=dead}}&lt;/ref&gt;
A confrontation between Adonis &quot;Donnie&quot; Creed and Ivan Drago was hinted at on [[Instagram]].&lt;ref&gt;{{cite web|url=https://www.yahoo.com/movies/sylvester-stallone-teasing-ivan-dragos-return-creed-2-135833267.html|title=Sylvester Stallone Teases an Ivan Drago Angle for 'Creed 2'|date=July 6, 2017 |access-date=July 7, 2017|archive-date=July 7,
```

```
2017|archive-url=https://web.archive.org/web/20170707233857/https://www.yahoo.com/movies/sylvester-stallone-teasing-ivan-dragos-return-creed-2-135833267.html|url-status=live}}&lt;/ref&gt; Stallone later confirmed that he had finished writing the script for the sequel, which would see [[Dolph Lundgren]] reprising his role as [[Ivan Drago]] from ''Rocky IV''.&lt;ref&gt;{{cite web|url=https://www.comingsoon.net/movies/news/872283-drago-confirmed-for-creed-ii-as-stallone-finishes-script#/slide/1|title=Drago Confirmed for Creed II as Stallone Finishes Script|date=July 21, 2017|access-date=April 17, 2020|archive-date=November 9, 2020|archive-url=https://web.archive.org/web/20201109014836/https://www.comingsoon.net/movies/news/872283-drago-confirmed-for-creed-ii-as-stallone-finishes-script#/slide/1|url-status=live}}&lt;/ref&gt; By October 2017, Stallone stated on his social media page that he would direct the film,&lt;ref&gt;{{cite news|url=https://www.slashfilm.com/the-creed-2-director-is-sylvester-stallone/|title='Creed 2' Will Return Sylvester Stallone to the Director's Chair|last=Hall|first=Jacob|publisher=[[/Film]]|date=October 9, 2017|access-date=December 11, 2017|archive-date=November 28, 2017|archive-url=https://web.archive.org/web/20171128031328/http://www.slashfilm.com/the-creed-2-director-is-sylvester-stallone|url-status=live}}&lt;/ref&gt; however, by December of the same year, [[Steven Caple Jr.]] was announced as the director of ''Creed II''.&lt;ref&gt;{{cite news|url=https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/|title=Steven Caple Jr. to Direct 'Creed 2' Starring Michael B. Jordan and Sylvester Stallone|last=Kroll|first=Justin|work=[[Variety (magazine)|Variety]]|date=December 11, 2017|access-date=December 11, 2017|archive-date=April 12, 2019|archive-url=https://web.archive.org/web/20190412030538/https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/|url-status=live}}&lt;/ref&gt; Boxer Florian Munteanu was cast as Drago's son.&lt;ref&gt;{{cite magazine|url=https://ew.com/movies/2018/01/16/creed-2-ivan-drago-son-florian-munteanu/|title=Creed 2 casts pro boxer as Ivan Drago's son|last=Romano|first=Nick|magazine=[[Entertainment Weekly]]|date=January 16, 2018|access-date=January 30, 2018|archive-date=January 31, 2018|archive-url=https://web.archive.org/web/20180131051028/http://ew.com/movies/2018/01/16/creed-2-ivan-drago-son-florian-munteanu/|url-status=live}}&lt;/ref&gt; In an interview on ''[[The Ellen DeGeneres Show]]'', Jordan confirmed that ''Creed II'' was his next project.&lt;ref&gt;{{Cite episode |title=Michael B. Jordan |series=Ellen: The Ellen DeGeneres Show |first= Ellen (host) |last= DeGeneres |date= 6 February 2018 |season=15 |number=97}}&lt;/ref&gt;
2756 {{Sylvester Stallone}}
2757 Moon joined [[Seoul SK Knights]] in 2006. He retired at the end of the 2009-10 season.&lt;ref name=&quot;ChosunIlbo.May2010&quot;/&gt; The team retired his number 10 jersey.&lt;ref&gt;{{cite news|title=&lt;농구소식&gt; SK, 문경은 '10 번' 영구결번|url=https://www.yna.co.kr/view/AKR20101012042000007|work=[[Yonhap News Agency]]|date=October 12, 2010|language=ko}}&lt;/ref&gt;&lt;ref name=&quot;joins.May2018&quot;/&gt; Known for being one of the best three-point shooters during his heyday, he earned the nickname &quot;[[John Rambo|Rambo]] Shooter&quot; as he was said to resemble Rambo actor [[Sylvester Stallone]].&lt;ref name=&quot;joins.May2018&quot;&gt;{{Cite web|url=http://news.jtbc.joins.com/article/article.aspx?news_id=NB11638778|title=[IS 화보스토리]서울 SK 문경은 감독. 이제 다음을 준비해야 할때.|publisher=[[JTBC]]|date=May 22, 2018|language=ko|access-date=June 15, 2018}}&lt;/ref&gt;&lt;ref name=&quot;joins.Oct2020&quot;/&gt;
2758 In October 2015, there were reports of [[American cinema|American]] actor [[Sylvester Stallone]] being cast in the film as Khan's on-screen coach.&lt;ref&gt;{{cite web | url=http://zeenews.india.com/entertainment/movies/sylvester-stallone-to-star-in-salman-khans-sultan_1811613.html | title=Sylvester Stallone to star in Salman Khan's 'Sultan'? | publisher=[[Zee News]] | work=[[Zee Entertainment Enterprises]] | date=18 October 2015 | access-date=16 December 2015}}&lt;/ref&gt; However, Raja Mukerji, the Executive Producer of the film cleared that Stallone would not be starring in the film.&lt;ref&gt;{{cite web | url=http://indianexpress.com/article/entertainment/bollywood/sylvester-stallone-is-not-with-salman-khan-in-sultan-confirms-executive-producer-raja-mukerji-rani/ | title=Sylvester Stallone is not in Salman Khan's Sultan, confirms film's executive producer Raja Mukerji | publisher=[[The Indian Express]] | date=27 October 2015 | access-date=16 December 2015 | last=Sonup | first=Sahadevan}}&lt;/ref&gt; Later, [[Sanjay Dutt]] was then rumoured to play the role of coach according to Khan's insistence.&lt;ref&gt;{{cite web | url=http://www.dnaindia.com/entertainment/report-sanjay-dutt-to-play-salman-khan-s-coach-in-sultan-2156807 | title=Sanjay Dutt to play Salman Khan's coach in 'Sultan'?| publisher=[[DNA India]] | work=DNA desk | date=17 December 2015 | access-date=16 December 2015}}&lt;/ref&gt; However, Khan later reported that he never insisted Aditya Chopra for Dutt's role.{{cn|date=September 2024}}
2759 | colspan=&quot;4&quot; | [[Sylvester Stallone]]
2760 Barney Ross, portrayed by [[Sylvester Stallone]], is the leader of the Expendables. He is hired by Agent Church in the first film to investigate suspicious activities on the island of Vilena and take out its fanatical dictator, General Garza. Barney considers abandoning the job after finding out that renegade CIA agent James Munroe is using Vilena as a source of narcotics production. However, he is impressed by the courage of Garza's rebellious daughter and ends up taking down both Garza and Munroe. Though Barney completes his assignment, Church wants compensation for the death of Munroe and hires the Expendables in [[The Expendables 2|the second film]] to recover a briefcase containing sensitive information from a downed plane in Albania . Since the case is locked in a booby-trapped safe, Church forces Barney to take hacker Maggie Chan along for the mission. Barney reluctantly accepts and the mission goes off without a hitch. However, Barney's crew is ambushed by arms-dealer Jean Vilain and his gang, the Sangs. Vilain murders Barney's rookie crew member Billy the Kid and takes the briefcase, which contains a computer map revealing the location of five tons of discarded plutonium. Vowing vengeance, Barney and the crew hunt down Vilain. After a one-on-one battle at an airport, Barney kills Vilain.
2761 The film moved swiftly into production, within a week of Amir Shervan and ex-[[Sylvester Stallone]] bodyguard&lt;ref&gt;[https://drafthouse.com/event/psycho-cinema-samurai-cop Alamo Dafthouse Cinema]&lt;/ref&gt; Matt Hannon's first meeting. Upon walking into Shervan's office for the first time, Matt Hannon was told that he was perfect and was immediately handed the full script to the film. Despite the film being titled ''Samurai Cop'', Hannon had no experience with weapons training, and all of his formal practice would be classified as [[mixed martial arts|MMA]]. As a result of Hannon's lack of experience and Shervan's inability to direct, all of the combat scenes were choreographed by martial arts expert Gerald Okamura, or were planned out by the actors themselves, sometimes only fifteen minutes before the scene in question was filmed.
2762 Mathew Karedas was one of [[Sylvester Stallone]]'s bodyguards during the late 1980s. This subsequently inspired him to pursue acting. &quot;In Sly's world, it's a very rarefied air for the very wealthy - it's intoxicating. At that time, he'd just finished ''[[Rambo III]]'' and ''[[Tango &amp; Cash]]'' and spending all that time hanging out with all these celebrities, I decided I wanted to have at least 1% of what he had!&quot;&lt;ref name=&quot;Bristol&quot;&gt;{{Cite web|date=2015-09-23|title=EXCLUSIVE INTERVIEW: Matt Karedas (aka Matt Hannon) - star of SAMURAI COP and SAMURAI COP 2: DEADLY VENGEANCE|url=https://bristolbadfilmclub.co.uk/exclusive-interview-matt-karedas-aka-matt-hannon-star-of-samurai-cop-and-samurai-cop-2-deadly-vengeance/|access-date=2021-07-17|website=Bristol Bad Film Club|language=en-GB}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|title=My film is so bad, it's good|url=https://www.oldham-chronicle.co.uk/news-features/101/features/96822/my-film-is-so-bad-it's-good|access-date=2021-07-17|website=www.oldham-chronicle.co.uk}}&lt;/ref&gt;
2763 In June 2020 Charlotte was credited as Co-Executive Producer of ''Becoming Rocky'' (international title ''40 Years of Rocky'') released by her company Branded Studios&lt;ref&gt;{{cite web|url=https://www.prnewswire.co.uk/news-releases/sylvester-stallone-s-rocky-documentary-secures-worldwide-release-as-becoming-rocky--864212801.html|title=Sylvester Stallone's Rocky Documentary Secures Worldwide Release as 'Becoming Rocky'|first=PR|last=Newswire|date=6 June 2020|publisher=|accessdate=5 June 2020|via=PR Newswire}}&lt;/ref&gt; the official documentary on the creation of Rocky, narrated by [[Sylvester Stallone]].&lt;ref&gt;{{cite web|url=https://eu.usatoday.com/story/entertainment/movies/2020/06/08/narrator-sylvester-stallone-best-revelations-40-years-rocky-new-documentary/3150268001/|title=Narrator Sylvester Stallone Best Revelations 40 Years of Rocky|first=Branded|last=Studios|date=6 June 2020|publisher=|accessdate=6 June 2020|via=Branded Studios}}&lt;/ref&gt; She is also credited as Co-Executive Producer on the 2021 documentary ''Stallone, Frank That Is'' on [[Frank Stallone]], Sylvester's younger brother. &lt;ref&gt;{{cite web|url=https://www.comingsoon.net/movies/trailers/1156732-exclusive-stallone-frank-that-is-trailer-from-branded-studios-new-documentary|title=Exclusive Stallone Frank That Is Trailer|first=Coming|last=Soon|date=2 December 2020|publisher=|accessdate=2 December 2020|via=Coming Soon}}&lt;/ref&gt;
2764 In March 2019, The Boies/Schiller Film Group filed suit on Tuesday, accusing Emmett/Furla Oasis Films of breaching a deal to produce two ''Escape Plan'' sequels starring Sylvester Stallone. According to the suit, Boies/Schiller put up a $6 million loan for the $65 million project. The arrangement also gave Boies/Schiller rights to finance and produce the sequels, ''Escape Plan 2: Hades'' and ''Escape Plan: The Extractors''. Boies/Schiller alleges that Emmett/Furla Oasis Films breached that agreement, and did not allow Boies/Schiller to put up financing for the sequels. The suit claims Emmett/Furla also failed to pay an executive producing fee on each of the two movies.&lt;ref&gt;{{Cite news|url=https://variety.com/2019/film/news/boies-schiller-escape-plan-lawsuit-emmett-furla-1203178856/|title=Boies/Schiller Film Group Sues Over 'Escape Plan' Sequels |last=Maddaus |first=Gene |date=2019-04-02|work=Variety|language=en|access-date=2019-07-20}}&lt;/ref&gt;
2765 *''[[Nighthawks (1981 film)|Nighthawks]]'' (1984 Fuji TV edition) - Deke DaSilva ([[Sylvester Stallone]])
2766  |Guests           = [[Michael Caine]], [[Sylvester Stallone]], [[Michael B. Jordan]] and [[Sue Perkins]]
2767  |Guests           = [[Rio Ferdinand]], [[Martin Freeman]], [[Tyson Fury]], [[Danai Gurira]], [[Ellie Taylor]], [[Sylvester Stallone]], and [[Little Simz]]
2768 | ''[[Ratchet &amp; Clank (film)|Ratchet &amp; Clank]]'' || [[Gramercy Pictures]] / [[PlayStation Productions|PlayStation Originals]] / [[Rainmaker Entertainment]] / Blockade Entertainment || [[Kevin Munroe]] (director); T.J. Fixman, Kevin Munroe, Gerry Swallow (screenplay); [[Paul Giamatti]], [[John Goodman]], [[Bella Thorne]], [[Rosario Dawson]], [[Jim Ward (voice actor)|Jim Ward]], [[Armin Shimerman]], [[Vincent Tong (voice actor)|Vincent Tong]], Andrew Cownden, [[James Arnold Taylor]], [[David Kaye (voice actor)|David Kaye]], [[Sylvester Stallone]] || &lt;ref&gt;{{Cite web|last=Ford|first=Rebecca|date=2015-05-13|title=Cannes: Focus Nabs Video Game Movie 'Ratchet &amp; Clank' for U.S. (Exclusive)|url=https://www.hollywoodreporter.com/movies/movie-news/cannes-focus-nabs-video-game-795274/|access-date=2021-07-18|website=The Hollywood Reporter|language=en-US}}&lt;/ref&gt;
2769 Tweed has represented [[Liam Neeson]], [[Britney Spears]],&lt;ref&gt;{{Cite web|url=https://abcnews.go.com/Video/playerIndex?id=2452034|title=Video|website=[[ABC News (United States)|ABC News]]}}&lt;/ref&gt; [[Jennifer Lopez]], [[Justin Timberlake]],&lt;ref&gt;{{Cite web|url=https://www.theguardian.com/media/2014/oct/21/justin-timberlake-jessica-biel-heat-lawsuit|title = Justin Timberlake and Jessica Biel settle Heat lawsuit|website = [[TheGuardian.com]]|date = 21 October 2014}}&lt;/ref&gt; [[Kelsey Grammer]], [[Nicolas Cage]],&lt;ref&gt;{{cite news|url=https://www.bbc.co.uk/news/uk-northern-ireland-19803213 | publisher=BBC News | title=Nicolas Cage receives damages and apology over online story | date=2 October 2012}}&lt;/ref&gt; [[Harrison Ford]], [[Chris de Burgh]], [[Neil Jordan]], [[Uri Geller]],&lt;ref&gt;{{cite news|url=https://www.bbc.co.uk/news/uk-northern-ireland-12964841|title=BBC News Libel lawyer Paul Tweed's work for Uri Geller on TV|publisher=BBC News | date=5 April 2011}}&lt;/ref&gt; [[Patrick Kielty]], [[Colin Farrell]], [[Keith Duffy]], [[The Corrs]], [[Ashton Kutcher]], [[Johnny Depp]] and [[Sylvester Stallone]].
2770 Brooker served on the Philadelphia Art Commission during two controversial periods. He was a member of the commission in 2006 when it was considering whether to return the “Rocky” statue to the steps of the Philadelphia Museum of Art. Brooker opposed the move. He labeled the statue - which was used in Sylvester Stallone's 1976 “[[Rocky|Rocky” movie]] - a prop and not art. The commission voted to place it in a spot near the steps of the museum.&lt;ref&gt;{{Cite news|date=2006-09-07|title=Philadelphia statue returning to museum spot|work=Journal Times (Racine, WS)|agency=via newspapers.com.|url=https://www.newspapers.com/image/343505012/?terms=%22moe%20brooker%22|access-date=2022-02-04}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|date=2006-08-05|title=Is Rocky art? Debate leaves him in limbo|work=Cincinnati Enquirer|agency=via newspapers.com.|url=https://www.newspapers.com/image/103553457/?terms=%22moe%20brooker%22|access-date=2022-02-04}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|last=Loviglio|first=Joann|date=2006-09-07|title=Philly hails 'cultural icon'|work=Ottawa Citizen (Ontario, Canada)|agency=via newspapers.com.|url=https://www.newspapers.com/image/466889246/?terms=%22moe%20brooker%22|access-date=2022-02-04}}&lt;/ref&gt;
2771 * [[Sylvester Stallone]] (1992)
2772 A former cop, Dash began working as a stuntman and actor beginning in 1977. He had small roles in ''[[Wolfen (film)|Wolfen]]'' and ''[[The Jazz Singer (1980 film)|The Jazz Singer]]''. That led to work on [[Sylvester Stallone]]'s ''[[Nighthawks (1981 film)|Nighthawks]],'' where he met Cliff Cudney. Cudney hired him to replace Warrington Gillette, who was originally scheduled to play Jason in ''[[Friday the 13th Part 2]]''.&lt;ref&gt;{{Cite web|url=http://fridaythe13thfilms.com/f13-hall-of-fame-steve-daskawisz/|title=F13 Hall of Fame: Steve Daskawisz|last=Sellers|first=Christian|date=January 12, 2011|website=FridayThe13thFilms|access-date=December 21, 2018}}&lt;/ref&gt;&lt;ref&gt;{{cite news|url=http://www.wpbf.com/entertainment/jason-vorhees-now-drives-cab-in-west-palm-
```



```
     beach/22720242|title=Jason Vorhees now drives cab in West Palm Beach|last=Guy|first=Erin|date=October 31, 2013|work=[[WPBF]]|accessdate=May 14, 2016}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.dreadcentral.com/news/3379/dash-steve-friday-the-13th-part-2/|title=Dash, Steve (Friday the 13th Part 2)|last=Condit|first=Jon|date=July 1, 2006|work=[[Dread Central]]|accessdate=May 14, 2016}}&lt;/ref&gt; Daskewisz died on December 18, 2018, at the age of 74 due to [[diabetes]]-related complications.&lt;ref&gt;{{Cite web|url=https://bloody-disgusting.com/news/3538857/r-p-jason-voorhees-friday-13th-part-2-steve-dash-passed-away/|title=[R.I.P.] Jason Voorhees from 'Friday the 13th Part 2,' Steve Dash Has Passed Away|last=Squires|first=John|date=December 19, 2018|website=Bloody Disgusting|access-date=December 21, 2018}}&lt;/ref&gt;
2773 The winners were announced during the awards ceremony on February 28, 2016.&lt;ref&gt;{{cite news|last1=Phillips |first1=Michael |title='Spotlight' takes best picture, DiCaprio wins first Oscar |url=https://www.chicagotribune.com/entertainment/movies/ct-oscars-2016-20160228-story.html |access-date=July 14, 2016 |work=[[Chicago Tribune]] |date=February 28, 2016 |url-status=live |archive-url=https://web.archive.org/web/20160711093318/http://www.chicagotribune.com/entertainment/movies/ct-oscars-2016-20160228-story.html |archive-date=July 11, 2016}}&lt;/ref&gt; With two Oscars, ''Spotlight'' was the first film since 1952's ''[[The Greatest Show on Earth (film)|The Greatest Show on Earth]]'' to win Best Picture with only one other award.&lt;ref&gt;{{cite news|title='Spotlight' Wins Oscar For Best Picture; Pope Challenged By Producer From Stage |url=https://deadline.com/2016/02/spotlight-oscar-winner-best-picture-2016-academy-award-tom-mccarthy-1201710767/ |first=Donald |last=Clarke |website=Deadline Hollywood |date=February 28, 2016 |access-date=February 29, 2016 |url-status=live |archive-url=https://web.archive.org/web/20160301092119/http://deadline.com/2016/02/spotlight-oscar-winner-best-picture-2016-academy-award-tom-mccarthy-1201710767/ |archive-date=March 1, 2016}}&lt;/ref&gt; [[Alejandro González Iñárritu|Alejandro G. Iñárritu]] became the third individual to win two consecutive Oscars for [[Academy Award for Best Director|Best Director]].{{refn|name=Consecutive win for Director|group=N|The two previous directors to have done so are: [[John Ford]] and [[Joseph L. Mankiewicz]]&lt;ref name=&quot;Spotlight&quot;/&gt;}}&lt;ref name=&quot;Spotlight&quot;/&gt; By virtue of his previous nomination for his portrayal of the [[Rocky Balboa|titular character]] in 1976's ''[[Rocky]]'', Best Supporting Actor nominee [[Sylvester Stallone]] was the sixth person to be nominated for playing the same role in two different films.&lt;ref&gt;{{cite news|title=Oscars 2016: Sylvester Stallone is thrilled to be back in Oscar ring after 39 years |url=https://www.latimes.com/entertainment/tv/showtracker/la-et-mn-oscars-2016-sylvester-stallone-returns-to-oscar-ring-after-39-years-20160114-story.html |date=January 14, 2016 |work=Los Angeles Times |first=Susan |last=King |access-date=March 24, 2016 |url-status=live |archive-url=https://web.archive.org/web/20160401224137/http://www.latimes.com/entertainment/tv/showtracker/la-et-mn-oscars-2016-sylvester-stallone-returns-to-oscar-ring-after-39-years-20160114-story.html |archive-date=April 1, 2016}}&lt;/ref&gt;
2774 ** [[Sylvester Stallone]]&amp;nbsp;- ''[[Creed (film)|Creed]]'' as [[Rocky Balboa]]
2775 |&lt;ref&gt;[https://deadline.com/2016/03/strong-premiere-date-sylvester-stallone-the-voice-nbc-1201712289/ &quot;Sylvester Stallone-Produced NBC Reality Series 'Strong' Gets Thursday Slot, Preview Behind 'The Voice'&quot;] from Deadline, 3/1/2016&lt;/ref&gt;
2776 Ke Jinde and Dr Glen Donnar delve into the resurgence of aging action stars like [[Arnold Schwarzenegger]], [[Sylvester Stallone]], and [[Bruce Willis]], reviving '80s action franchises characterized by simplistic, violence-driven portrayals of hyper-masculinity.{{sfn|Jinde|2022|pp=8-9}}{{sfn|Donnar|2021|p=243}} Jinde observes the evolution in representations of aging heroes, especially Ethan in the ''Mission: Impossible'' series. The modern aging hero focuses on social connections, self-sacrifice, and action beyond violence, reshaping the perception of mature heroes.{{sfn|Jinde|2022|p=18}} This transformation involves sacrifices such as Ethan's choice to end his relationship with Julia and forgo a settled family life, driven by his sense of responsibility to the world.{{sfn|Jinde|2022|p=16}}
2777 The film begins with Brian Thompson's explanation that the movie is his answer to the question: &quot;What's it like to work with [[Sylvester Stallone]], [[Arnold Schwarzenegger]], [[Steven Seagal]] or [[Jean-Claude Van Damme]]?&quot;  By combining the distinguishable features of these movie [[Action hero|stars]], he created one fictional character, Vardell Duseldorfer or VD. VD is an aged and retired action movie star who is now a drug addict and alcoholic. VD plans to return to fame through directing and starring in a science fiction movie called ''Hard Times on Mars''.&lt;ref name=&quot;:0&quot;&gt;{{Cite web|title=The Extendables (2014)|url=https://www.imdb.com/title/tt0996664/|access-date=2021-07-07|website=IMDb|language=en-US}}&lt;/ref&gt;
2778 | [[Sylvester Stallone]]{{ref|voice|V}}&lt;ref name=&quot;StalloneTheSS&quot; /&gt;
2779 &lt;ref name=&quot;StalloneTheSS&quot;&gt;{{Cite web |last=Chitwood |first=Adam |date=March 26, 2021 |title=Surprise! Sylvester Stallone Is King Shark in 'The Suicide Squad' |url=https://collider.com/the-suicide-squad-king-shark-actor-sylvester-stallone/ |url-status=live |archive-url=https://web.archive.org/web/20210327000409/https://collider.com/the-suicide-squad-king-shark-actor-sylvester-stallone/ |archive-date=March 27, 2021 |access-date=April 8, 2021 |website=[[Collider (website)|Collider]]}}&lt;/ref&gt;
2780 |{{sort|Stallone|[[Sylvester Stallone]]}}
2781 | ''[[Animal Crackers (2017 film)|Animal Crackers]]'' || [[Netflix]] || [[Scott Christian Sava]] (director/screenplay); [[Tony Bancroft]] (director); [[Dean Lorey]] (screenplay); [[John Krasinski]], [[Emily Blunt]], [[Ian McKellen]], [[Danny DeVito]], [[Sylvester Stallone]], [[Raven-Symoné]], [[Patrick Warburton]] ||style=&quot;text-align:center;&quot;| &lt;ref&gt;{{cite web|title='Animal Crackers' Trailer: Netflix's Rescued Animated Feature Gets Summer Streaming Date|url=https://www.indiewire.com/2020/06/animal-crackers-trailer-netflix-rescued-animated-feature-stream-july-24-1202237271/|date=June 15, 2020|website=Indie Wire|access-date=July 28, 2020|archive-date=September 1, 2022|archive-url=https://web.archive.org/web/20220901154223/https://www.indiewire.com/2020/06/animal-crackers-trailer-netflix-rescued-animated-feature-stream-july-24-1202237271/|url-status=live}}&lt;/ref&gt;
2782 Bleifer continued working through the 1960s, 1970s, and into the 1980s.  He made numerous television appearances on shows such as ''[[Death Valley Days]]'' (1960), ''[[The Lawless Years]]'' (1961), ''[[The Untouchables (1959 TV series)|The Untouchables]]'' (1961), ''[[Dr. Kildare (TV series)|Dr. Kildare]]'' (1962-3), ''[[The Man from U.N.C.L.E.]]'' (1966), ''[[Marcus Welby, M.D.]]'' (1969), ''[[Adam-12]]'' (1971), ''[[Kolchak: The Night Stalker]]'' (1974), ''[[S.W.A.T. (1975 TV series)|S.W.A.T.]]'' (1975), ''[[Police Woman (TV series)|Police Woman]]'' (1976), ''[[Little House on the Prairie (TV series)|Little House on the Prairie]]'' (1979), ''[[The White Shadow (TV series)|The White Shadow]]'' (1979), and ''[[Highway to Heaven]]'' (1984). Bleifer also appeared in the television miniseries ''[[QB VII#Television miniseries|QB VII]]'', in the role of Ben-Dan.{{cn|date=November 2019}} While his activity in films decreased, he did continue in the medium, with roles in such films as: the tobacconist in the 1962 comedy ''[[If a Man Answers]]'', starring [[Bobby Darrin]] and [[Sandra Dee]];&lt;ref&gt;{{cite web | url=http://www.afi.com/members/catalog/DetailView.aspx?s=&amp;Movie=22870 | title=If a Man Answers: Detail View | publisher=American Film Institute | accessdate= July 26, 2015}}&lt;/ref&gt; the Steward in ''[[The Hook (1963 film)|The Hook]]'', starring Kirk Douglas;&lt;ref&gt;{{cite web | url=http://www.afi.com/members/catalog/DetailView.aspx?s=&amp;Movie=22416 | title=The Hook: Detail View | publisher=American Film Institute | accessdate= July 26, 2015}}&lt;/ref&gt; a small role in ''[[W.C. Fields and Me]]'' (1976), starring [[Rod Steiger]] and [[Valerie Perrine]];&lt;ref&gt;{{cite web | url=http://www.afi.com/members/catalog/DetailView.aspx?s=&amp;Movie=55654 | title=W.C. Fields and Me: Detail View | publisher=American Film Institute | accessdate= July 26, 2015}}&lt;/ref&gt; as Mishka in ''[[F.I.S.T. (film)|F.I.S.T.]]'' (1978), starring [[Sylvester Stallone]], Rod Steiger, and [[Peter Boyle]];&lt;ref&gt;{{cite web | url=http://www.afi.com/members/catalog/DetailView.aspx?s=&amp;Movie=55253 | title=F.I.S.T.: Detail View | publisher=American Film Institute | accessdate= July 26, 2015}}&lt;/ref&gt; and as one of the Rabbis in the [[Robert Aldrich]] comedy, ''[[The Frisco Kid]]'', starring [[Gene Wilder]] and [[Harrison Ford]].&lt;ref&gt;{{cite web | url=http://www.afi.com/members/catalog/DetailView.aspx?s=&amp;Movie=56771 | title=The Frisco Kid: Detail View | publisher=American Film Institute | accessdate= July 26, 2015}}&lt;/ref&gt; Bleifer's final performance was in the featured role of Hyman in 1986's ''Inside Out'', starring [[Elliott Gould]].{{cn|date=November 2019}} Bleifer died on January 24, 1992, in Los Angeles County, California, and was interred in the [[Hillside Memorial Park]], in Culver City, California, next to his wife, Grace, who had died three years previously, in 1989.{{Citation needed |date=March 2023}}
2783 As a Hollywood screenwriter, Kamen was mentored by [[Frank Price]] who told him that producer [[Jerry Weintraub]] had optioned a news article about the young child of a single mother who had earned a black belt to defend himself against the neighborhood bullies. Kamen then combined his own life story with the news article and used both to create the screenplay for ''The Karate Kid''.&lt;ref name=oralhistory/&gt; Additionally, given [[John G. Avildsen]]'s involvement with both films, [[Sylvester Stallone]] often joked with Kamen that the writer had &quot;ripped off&quot; the ''[[Rocky (film series)|Rocky]]'' films with ''The Karate Kid''.&lt;ref name=oralhistory/&gt;
2784 In 1990, Steiger starred in ''[[Men of Respect]]'', an adaptation of [[William Shakespeare]]'s play ''[[Macbeth]]''. Steiger's role in the critically panned thriller opposite [[Sylvester Stallone]] ''[[The Specialist]]'' (1994) earned him a nomination for the [[Golden Raspberry Award for Worst Supporting Actor]].&lt;ref&gt;{{cite web | url=http://www.razzies.com/forum/1994-razzie-nominees-winners_topic344.html | title=Fifteenth Annual Razzie Awards (for 1994) | publisher=[[Golden Raspberry Awards]] | date=December 4, 2005 | accessdate=January 25, 2016}}&lt;/ref&gt;&lt;ref&gt;{{cite book|last=Wilson|first=John|title=The Official Razzie Movie Guide: Enjoying the Best of Hollywood's Worst |year=2005|publisher=Grand Central Publishing|isbn=0-446-69334-0}}&lt;/ref&gt; Steiger appeared in both ''[[Shiloh (film)|Shiloh]]'' (1995) and [[Shiloh 2: Shiloh Season|the sequel]], three years later. He reunited with Jewison on the biopic ''[[The Hurricane (1999 film)|The Hurricane]]'' in 1999, in which he portrayed judge [[H. Lee Sarokin]]. He was one of the leads in the drama film ''[[The Last Producer]]'' (2000), before starring in his final film, the drama thriller ''[[Poolhall Junkies]]'' (2002).
2785 [[Clint Eastwood]] celebrated his 1992 Oscar success at the restaurant.&lt;ref name=&quot;James2014&quot;&gt;{{cite book|last=James|first=Anthony|title=Acting My Face: A Memoir|url=https://books.google.com/books?id=arLWAgAAQBAJ&amp;pg=PA158|date=6 February 2014|publisher=Univ. Press of Mississippi|isbn=978-1-61703-985-0|page=158}}&lt;/ref&gt; [[Sylvester Stallone]] also frequented Nicky Blair's and would take his dates there.&lt;ref name=&quot;Kaye2015&quot;&gt;{{cite book|last=Kaye|first=Elizabeth|title=Men: What They Do, What They Think, and Why ...|url=https://books.google.com/books?id=w7hICAAAQBAJ&amp;pg=PT183|date=21 April 2015|publisher=Byliner|isbn=978-1-5080-1282-5|page=183}}&lt;/ref&gt;
2786 | quote  = &quot;I knew it was going to be a hit. [[Sylvester Stallone]] was in the studio when we recorded it. If you listen very close to the original recording, you can hear in the mix the sound of him hitting the small bag.&quot;
2787 | source = -Maynard Ferguson describing Sylvester Stallone punching a speed bag to the rhythm of the song as it was recorded.&lt;ref&gt;{{cite web
2788 Covington co-wrote the screenplay for the 2015 film ''[[Creed (film)|Creed]]'', a [[Spin-off (media)|spin-off-sequel]] to the [[Rocky (film series)|''Rocky'' film series]], starring [[Sylvester Stallone]] and [[Michael B. Jordan]], with [[Ryan Coogler]], who  also directed the film.&lt;ref&gt;{{cite web |url=https://www.slashfilm.com/creed-synopsis-says-rocky-balboa-has-his-own-battle-in-the-new-film/|title='Creed' Synopsis Says Rocky Balboa Has His Own Battle in the New Film|last1=Fischer|first1=Russ|date=February 25, 2015|website=slashfilm.com|publisher=[[/Film|Slashfilm]]|access-date=June 30, 2015|quote=We'll find out when Creed opens on November 25. [[Ryan Coogler]] ([[Fruitvale Station|Fruitvale]]) directs from a script he wrote with Aaron Covington.}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=http://www.digitaltrends.com/movies/sylvester-stallone-rocky-spinoff-creed-photo/|title=Rocky Balboa is back in the ring in first photo from spinoff, Creed|last1=Marshall|first1=Rick |date=April 14, 2015|website=digitaltrends.com|publisher=[[Digital Trends]]|access-date=June 30, 2015|quote=The film is based on a script penned by [[Ryan Coogler|Coogler]] and Aaron Covington, and was shot on location in [[Philadelphia]], [[Rocky Balboa|Rocky's]] hometown.}}&lt;/ref&gt;&lt;ref name=&quot;:0&quot; /&gt; Covington is a personal friend of Coogler,&lt;ref&gt;{{cite web |url=https://blogs.wsj.com/speakeasy/2013/12/03/ryan-coogler-on-writing-creed-and-resurrecting-rocky-balboa/|title=Ryan Coogler on Writing 'Creed' and Resurrecting Rocky Balboa|last1=Cheney|first1=Alexandra|date=December 3, 2013|website=blogs.wsj.com|publisher=[[The Wall Street Journal]]|access-date=June 30, 2015|quote=Coogler met Stallone prior to production on &quot;Fruitvale&quot; and had &quot;a really honest conversation about these two men,&quot; said Coogler, who is co-writing &quot;Creed&quot; with a screenwriting friend, Aaron Covington. As soon as Stallone signed on, the two {{sic|deled}} deeper into developing the script, which is still being written.}}&lt;/ref&gt; and the two worked together with Stallone to [[green-light]] the film with [[Metro-Goldwyn-Mayer|MGM]] studios.&lt;ref&gt;{{cite web |url=https://deadline.com/2014/12/tessa-thompson-creed-sylvester-stallone-dear-white-people-selma-1201327391/|title='Dear White People's Tessa Thompson Lands 'Creed' Lead|last1=Yamato|first1=Jen|date=December 16, 2014|website=[[Deadline Hollywood]]|access-date=June 30, 2015|quote=Coogler, who scripted with Aaron Covington, teamed with Stallone to pitch the project to MGM and is helming it as his follow-up to his Cannes-, Gotham- and Spirit Award-winning 2013 debut Fruitvale Station.}}&lt;/ref&gt;&lt;ref
```

name=&quot;Breakout&quot;&gt;{{cite web |url=https://deadline.com/2013/07/fruitvale-station-duo-ryan-coogler-and-michael-b-jordon-team-with-sly-stallone-on-mgm-rocky-spinoff-creed-547901/|title='Fruitvale Station' Duo Ryan Coogler And Michael B. Jordan Team With Sly Stallone on MGM 'Rocky' Spinoff 'Creed'|last1=Fleming|first1=Mike Jr. |date=July 24, 2013|website=Deadline Hollywood|access-date=June 30, 2015|quote=[[Metro-Goldwyn-Mayer|MGM]] is setting [[Ryan Coogler]] to direct ''Creed'', and the studio is in early talks with Coogler's ''[[Fruitvale Station]]'' star [[Michael B. Jordan]] to play the grandson of [[Apollo Creed]] in a continuation of the [[Rocky (film series)|''Rocky'' saga]] that Coogler is going to write with Aaron Covington.}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://variety.com/2013/film/news/fruitvale-station-rocky-creed-mgm-1200567421/|title='Fruitvale Station' Team Eyeing 'Rocky' Spin-Off 'Creed' With MGM|last1=Kroll|first1=Justin|date=July 24, 2013 |work=[[Variety (magazine)|Variety]] |access-date=June 30, 2015|quote=Sylvester Stallone is on board to reprise his role as Rocky Balboa, with Coogler penning the script along with Aaron Covington. Deadline Hollywood broke the news.}}&lt;/ref&gt;

2789 | ''[[Creed (film)|Creed]]'' || 2015 || While trying to follow in his late father's footsteps by training with Rocky Balboa ([[Sylvester Stallone]]), the estranged son of Apollo Creed, Adonis Johnson-Creed ([[Michael B. Jordan]]), falls in love with a musician with progressive [[hearing loss]] named Bianca Taylor (played by hearing actress [[Tessa Thompson]]).{{citation needed|date=September 2019}}

2790 In 1998, Racefab Inc. partnered with [[Champ Car|CART]] to construct a driving experience series called CART Driving 101. Designs included a single seat Champ Car and a tandem seat Champ Car for ride-alongs. After CART ended in 2004, Driving 101 became The [[Mario Andretti]] Racing Experience. You can drive one of these Indy Cars today as they travel around the US. In 2001, these cars went on to be used in the movie [[Driven (2001 film)|Driven]] starring [[Sylvester Stallone]].&lt;ref&gt;{{cite web|title=Mario Andretti Racing Experience|url=http://www.andrettiracing.com/|website=Andretti Racing}}&lt;/ref&gt;

2791 * 2001 Racefab Inc. built cars featured in Driven with Sylvester Stallone

2792 *''[[Over the Top (1987 film)|Over the Top]]'' (Netflix edition) (Lincoln Hawk ([[Sylvester Stallone]]))

2793 | [[Sylvester Stallone]]

2794 *Read my [[Sylvester Stallone]] related posts from elsewhere, so he followed me here.

2795 Although it's not my fault he does what he does, I want to apologize to Wikipedia staff for this guy being a nuisance here. He followed me here because he monitored all these things I posted on message boards about i.e. my former support workers, my beef with [[The Internet Movie Database]], over their moderators removing verifiable film credits on the page for Sylvester Stallone, and me using forums to discuss the ''[[Resident Evil]]'' games, which is why he mass registered at a forum I go on that doesn't have any moderators. Everybody there is getting sick of his spam, but he just jumps proxies and registers again anyway. He also sent people to my friend's house a few weeks ago, after he posted fake ads on a site we couldn't trace. Someone on a forum advised me to 100% ignore him and not mention him or react at all any more, as talking about him is giving him energy to keep doing this.[[User:PeterMan844|PeterMan844]] ([[User talk:PeterMan844|talk]]) 22:35, 5 September 2017 (UTC)

2796 **&quot;Steiger played … Sylvester Stallone plays&quot; - troubles with tenses

2797 | [[Sylvester Stallone]]

2798 '''''The Director's Chair''''' is an American [[talk show]] created and hosted by filmmaker [[Robert Rodriguez]]. It premiered on May 10, 2014, on [[El Rey (network)|El Rey]] and features Rodriguez interviewing other filmmakers about their filmmaking techniques and their careers.&lt;ref&gt;{{cite magazine|url=http://www.ew.com/article/2015/11/10/sylvester-stallone-directors-chair-robert-rodriguez|title=Robert Rodriguez sits down with Sylvester Stallone in El Rey's The Director's Chair - free stream and exclusive poster|magazine=[[Entertainment Weekly]]|publisher=}}&lt;/ref&gt;

2799 |Title   = [[Sylvester Stallone]]&lt;ref&gt;{{Cite web|url=http://www.indiewire.com/2015/11/watch-full-50-minute-episode-of-the-directors-chair-with-robert-rodriguez-and-sylvester-stallone-105976/|title = Watch: Full 50-Minute Episode of 'The Director's Chair' with Robert Rodriguez and Sylvester Stallone|date = 11 November 2015}}&lt;/ref&gt;

2800 |ShortSummary   = Robert Rodriguez interviews director Sylvester Stallone to discuss Sly's beginnings and inspirations, what it's like to write, direct and star in an action film and his return to directing with Rocky Balboa, Rambo, and The Expendables.

2801 Ingalls was best known for her role as Bunchie, a prostitute who comforts [[Sylvester Stallone]]'s character, Cosmo Carboni, in the 1978 drama, ''[[Paradise Alley]]''.&lt;ref name=ew/&gt;&lt;ref name=thr/&gt; She also appeared in ''The Man Who Would Not Die''&lt;ref&gt;{{cite news|title=(advertisement)|url=https://www.newspapers.com/clip/4108432/the_leavenworth_times/|agency=The Leavenworth Times|date=April 14, 1976|location=Kansas, Leavenworth|page=15|via = [[Newspapers.com]]|access-date = January 23, 2016}} {{Open access}}&lt;/ref&gt; in 1975, ''[[Deadly Force (film)|Deadly Force]]'' in 1983, and ''[[Lethal Weapon 4]]'' in 1998.&lt;ref name=ew/&gt;

2802 In 1978, Ingalls was named when [[Sylvester Stallone]] was sued for divorce by his wife, Sasha, &quot;who claimed the star of ''[[Rocky (film series)|Rocky]]'' used community funds for a vacation for himself and actress Joyce Ingalls.&quot;&lt;ref&gt;{{cite news|title=(photo caption)|url=https://www.newspapers.com/clip/4108904/the_ottawa_journal/|agency=The Ottawa Journal|date=March 18, 1978|location=Canada, Ottawa, Ontario|page=16|via = [[Newspapers.com]]|access-date = January 23, 2016}} {{Open access}}&lt;/ref&gt; Earlier that year, gossip columnist [[Liz Smith (journalist)|Liz Smith]] had written: &quot;Sylvester (&quot;Sly&quot;) Stallone and his ''Paradise Valley'' co-star Joyce Ingalls are what one might call a real hot item. Mrs. Stallone knows all about it and is definitely not happy.&quot;&lt;ref&gt;{{cite news|last1=Smith|first1=Liz|title=Quoth a Lady: We Talk Too Much|url=https://www.newspapers.com/clip/4108988/colorado_springs_gazettetelegraph/|agency=Colorado Springs Gazette-Telegraph|date=February 20, 1978|location=Colorado, Colorado Springs|page=18|via = [[Newspapers.com]]|access-date = January 23, 2016}} {{Open access}}&lt;/ref&gt;

2803 [[File:Sylvester Stallone 2012.jpg|right|thumb|150px|[[Sylvester Stallone]], Best Supporting Actor winner]]

2804 * '''[[Sylvester Stallone]] - ''[[Creed (film)|Creed]]'' as [[Rocky Balboa]]'''

2805 At the American box office, the film surpassed the lifetime business of Rajinikanth's previous film ''[[Lingaa]]'' (2014) in just two days.&lt;ref name=&quot;day2overseas&quot; /&gt; The film debuted at number 11 at the American box office, earning $4.09&amp;nbsp;million.&lt;ref name=&quot;overseas creed&quot;&gt;{{cite news |last=Chauhan |first=Gaurang |date=7 December 2018 |title=2.0 triumphs over Hollywood as Rajinikanth beats Sylvester Stallone at the overseas box office |work=Times Now |url=https://www.timesnownews.com/entertainment/box-office/article/2-0-triumphs-over-hollywood-as-rajinikanth-beats-sylvester-stallone-at-the-overseas-box-office/327379 |access-date=7 December 2018}}&lt;/ref&gt; The screen count was increased from 20 to 75 in Pakistan on its second day to meet the demand.&lt;ref name=&quot;day2overseas&quot; /&gt; 2.0 grossed US$14.75{{nbsp}}million ({{INR}}1.03{{nbsp}}billion) in first five days in overseas markets.&lt;ref name=&quot;2.0five&quot; /&gt; On its fourth day, a Sunday, business picked up over India, which led the film to accumulate an opening weekend collection of around {{INRConvert|400|c|year=2018|mode=historical}}, the highest amount earned by any film in the week of 29 November to 2 December, ahead of ''[[Fantastic Beasts: The Crimes of Grindelwald]]'' (2018).&lt;ref name=&quot;thenewsminute_84533&quot; /&gt;&lt;ref name=&quot;Box office worldwide day4&quot;&gt;{{cite web |date=3 December 2018 |title=2.0 box office collection: Rajinikanth-Akshay's film earns Rs 400 crore worldwide |url=https://indianexpress.com/article/entertainment/tamil/rajinikanth-2-0-collects-rs-400-crore-worldwide-5475561/ |work=The Indian Express}}&lt;/ref&gt; Collections in India, discounting overseas, were {{INRConvert|291|c|year=2018|mode=historical}} gross ({{INR}}229&amp;nbsp;crore nett) in all languages.&lt;ref name=&quot;boiday4&quot;&gt;{{cite web |date=3 December 2018 |title=2.0: All India - Overseas - Worldwide Update |url=https://boxofficeindia.com/report-details.php?articleid=4486 |access-date=3 December 2018 |publisher=Box office India}}&lt;/ref&gt; This pushed it ahead of ''Enthiran'', which was the previous highest-grossing film from [[Tamil cinema]] with an earning of {{INR}}205&amp;nbsp;crore nett in India.&lt;ref name=&quot;boiday4&quot; /&gt;{{efn|While the Telugu-Tamil bilinguals ''[[Baahubali: The Beginning]]'' (2015) and ''[[Baahubali 2: The Conclusion]]'' (2017) crossed ''Enthiran''{{'}}s earnings, they were produced in [[Telugu cinema]].}} In its four-day opening weekend, the film opened at number one at the [[United Arab Emirates]] box office, earning $2.5&amp;nbsp;million, ahead of ''[[Creed II]]'' (2018).&lt;ref name=&quot;overseas creed&quot; /&gt;

2806 Later retrospective reviews has been mostly negative from audiences and critics, citing its poor writing and dialogue, acting, and its effects. Some gave the benefit of the doubt because it was purposefully a student film, but some state their gripes about its late release for just capitalizing into Daisy Ridley's fame, similar to [[Sylvester Stallone]] for ''[[The Party at Kitty and Stud's]]'' when ''[[Rocky]]'' was a big hit at the time. Ridley was also viewed for her lackluster performance, but was also the only favorable part of the film. [[IGN]]'s Tom Jorgensen gave it a 2 stars out of 10, stating that &quot;its story is incomprehensible, it isn't scary at all, and the acting is wall-to-wall bad&quot; and cites that the film &quot;is damn near unwatchable.&quot;&lt;ref&gt;{{Cite web |url=https://www.ign.com/articles/2019/06/27/scrawl-review-daisy-ridley-movie|title=Scrawl Review |work=[[IGN]] }}&lt;/ref&gt;

2807 |Guests=[[Sylvester Stallone]], [[Ted Koppel]]

2808 |ShortSummary=An appearance from [[2015 NASCAR Sprint Cup Series]] champion [[Kyle Busch]]. The ''Late Show'' Presents: Thanksgiving Turkey Tips with Our Friends at [[Butterball]]. Sylvester Stallone discusses ''[[Creed (film)|Creed]]'' and ''[[Rocky]]''. Ted Koppel discusses his new book, ''Lights Out''. My Morning Jacket performs &quot;Tropics&quot; from their album ''[[The Waterfall (album)|The Waterfall]]''.

2809 ** Runner-up: [[Sylvester Stallone]] - ''[[Creed (film)|Creed]]''

2810 *[[Sylvester Stallone]], ''[[Creed (film)|Creed]]''

2811 ** [[Sylvester Stallone]] - ''[[Creed (film)|Creed]]'' as [[Rocky Balboa]]

2812 | [[Sylvester Stallone]]

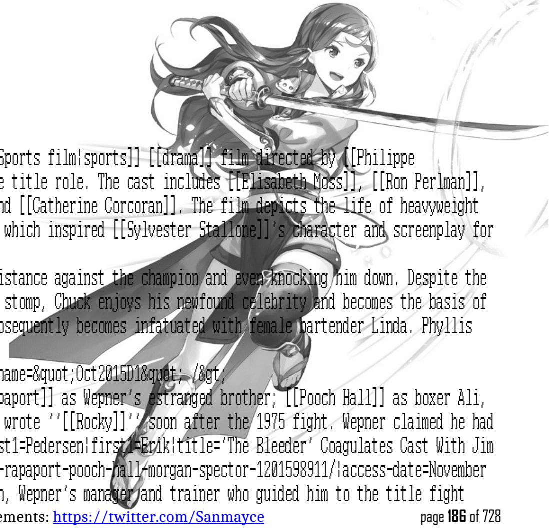
2813 | [[Sylvester Stallone]]&lt;br/&gt;[[Robert Pattinson]]&lt;br/&gt;[[Rachel Weisz]]

2814 '''''Chuck''''' (released as '''''The Bleeder''''' in the UK and Ireland) is a 2016 American [[Biographical film|biographical]] [[Sports film|sports]] [[drama]] film directed by [[Philippe Falardeau]] and written by [[Jeff Feuerzeig]], [[Jerry Stahl]], [[Michael Cristofer]] and [[Liev Schreiber]], who also stars in the title role. The cast includes [[Elisabeth Moss]], [[Ron Perlman]], [[Naomi Watts]], [[Jim Gaffigan]], [[Michael Rapaport]], [[Pooch Hall]], [[Morgan Spector]], [[Jason Jones (actor)|Jason Jones]] and [[Catherine Corcoran]]. The film depicts the life of heavyweight boxer [[Chuck Wepner]] and his [[Muhammad Ali vs. Chuck Wepner|1975 title fight]] with the heavyweight champion, [[Muhammad Ali]], which inspired [[Sylvester Stallone]]'s character and screenplay for the 1976 film ''[[Rocky]]''.

2815 Although the fight is largely one-sided, Chuck defies predictions of being knocked out in the third round, almost going the full distance against the champion and even knocking him down. Despite the loss, Wepner becomes a local hero. Though he is irritated to hear claims from Ali that Wepner's knockdown of him was due to a foot stomp, Chuck enjoys his newfound celebrity and becomes the basis of [[Sylvester Stallone]]'s film [[Rocky]]. However, Chuck becomes increasingly addicted to his fame and begins taking cocaine. He subsequently becomes infatuated with female bartender Linda. Phyllis soon catches on to Wepner's unfaithful behavior and kicks him out of the house.

2816 * [[Morgan Spector]] as [[Sylvester Stallone]], the actor-screenwriter who wrote ''[[Rocky]]'' right after the 1975 fight.&lt;ref name=&quot;Oct2015D1&quot; /&gt;

2817 On October 30, 2015, additional cast was announced, including [[Jim Gaffigan]] as John Stoehr, Wepner's loyal friend; [[Michael Rapaport]] as Wepner's estranged brother; [[Pooch Hall]] as boxer Ali, whose 1975 fight with Wepner led to Wepner's sudden fame; and [[Morgan Spector]] as actor-screenwriter [[Sylvester Stallone]], who wrote ''[[Rocky]]'' soon after the 1975 fight. Wepner claimed he had inspired that film's title character [[Rocky Balboa]] but Stallone has never confirmed it.&lt;ref name=Oct2015D1&gt;{{cite news|last1=Pedersen|first1=Erik|title='The Bleeder' Coagulates Cast With Jim Gaffigan, Michael Rapaport, Pooch Hall &amp; Morgan Spector|url=https://deadline.com/2015/10/the-bleeder-cast-jim-gaffigan-michael-rapaport-pooch-hall-morgan-spector-1201598911/|access-date=November 16, 2015|work=deadline.com|date=October 30, 2015}}&lt;/ref&gt; On November 11, 2015, [[Ron Perlman]] signed on to play Al Braverman, Wepner's manager and trainer who guided him to the title fight



with Ali.&lt;ref name=PerlmanCast&gt;{{cite news|last1=Fleming|first1=Mike Jr.|title=Ron Perlman Joins 'The Bleeder'|url=https://deadline.com/2015/11/ron-perlman-the-bleeder-chuck-wepner-liev-schreiber-1201618534/|access-date=November 16, 2015|work=deadline.com|date=November 11, 2015}}&lt;/ref&gt; On November 11, 2015, [[Remstar|Remstar Films]] acquired the Canadian distribution rights to the film.&lt;ref&gt;{{cite news|last1=A. Lincoln|first1=Ross|title='Headshot' Global Rights Nabbed By Nikkatsu, Vertical &amp; XYZ; Remstar Lands Canadian Rights To 'The Bleeder' - AFM|url=https://deadline.com/2015/11/headshot-the-bleeder-rights-acquired-afm-1201619113/|access-date=November 15, 2015|work=deadline.com|date=November 11, 2015}}&lt;/ref&gt;

2818 [[Category:Cultural depictions of Sylvester Stallone]]

2819 | [[Sylvester Stallone]]&lt;br/&gt;[[Tina Robertson]]

2820 *[[Sylvester Stallone]]

2821 As Irena Gerasimenko she worked as a model for ''[[Sports Illustrated]]''. Along with her roommate Bernadette Leonard (a fellow ''Sports Illustrated'' model) and [[Jennifer Flavin]] (a model and wife of [[Sylvester Stallone]]), she serves as Vice Chairman and collaborated to create a charity, C.O.A.C.H. for Kids and their Families which is a community outreach assistance for children's health.&lt;ref&gt;{{Cite web|title=Sugar Ray Leonard's Wife Bernadette Robi (Photos - Pictures)|url=http://ballerwives.com/2011/03/03/former-boxing-champion-sugar-ray-leonards-wife-bernadette-robi|website=BallerWives.com|access-date=2015-12-08|archive-date=2015-12-10|archive-url=https://web.archive.org/web/20151210212829/http://ballerwives.com/2011/03/03/former-boxing-champion-sugar-ray-leonards-wife-bernadette-robi/|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |url=https://www.cedars-sinai.edu/Community-Benefit/Programs-and--Services/COACH-for-Kids-and-Their-Families/Documents/PrivilegeArticle.pdf |title=Archived copy |access-date=2015-12-08 |archive-date=2015-10-02 |archive-url=https://web.archive.org/web/20151002105611/http://www.cedars-sinai.edu/Community-Benefit/Programs-and--Services/COACH-for-Kids-and-Their-Families/Documents/PrivilegeArticle.pdf |url-status=dead }}&lt;/ref&gt;&lt;ref&gt;{{Cite news|title = Fashioning a Benefit for Kids, Families|url=https://www.latimes.com/archives/la-xpm-1999-jan-20-cl-65210-story.html|newspaper=Los Angeles Times|date=1999-01-20|access-date=2015-12-06|issn=0458-3035|first=Irene|last=Lacher}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|title=Parties With a Purpose: Singers and Society Types Alike Help Charities|url=https://www.latimes.com/archives/la-xpm-2000-jan-04-cl-50423-story.html|newspaper=Los Angeles Times|date=2000-01-04|access-date=2015-12-08|issn=0458-3035|first=Patt|last=Diroll}}&lt;/ref&gt;

2822 Kabongo found early acting inspiration in [[action film]]s from the [[1990s in film|1990s]] that his mother showed him as a child.&lt;ref name=&quot;:4&quot;&gt;{{Cite web|date=August 28, 2017|title=How his mother's love for '90s action films inspired Emmanuel Kabongo's own acting career|url=https://www.cbc.ca/arts/exhibitionists/how-his-mother-s-love-for-90s-action-films-inspired-emmanuel-kabongo-s-own-acting-career-1.4265303|website=[[CBC Arts]]}}&lt;/ref&gt; In an interview with [[CBC Arts]], Kabongo said that &quot;taking acting serious came from watching how my mom reacted to films of the '90s ... [She] loves [[Denzel Washington|Denzel]], [[Jean-Claude Van Damme|Van Damme]], [[Sylvester Stallone]]... My mom watched all those movies growing up, and knowing how that made her feel, I just wanted to do the same thing for her.&quot;&lt;ref name=&quot;:4&quot; /&gt; Kabongo has cited [[Leonardo DiCaprio]], [[Tom Hardy]], [[Mahershala Ali]], [[Viola Davis]], and [[Daniel Day-Lewis]] as acting inspirations.&lt;ref name=&quot;:3&quot; /&gt;

2823 In 1998, Adonis &quot;Donnie&quot; Johnson is spending time at a juvenile detention center in [[Los Angeles]] and frequently getting into fights with other children. As a result, he is sent to solitary confinement. Mary Anne Creed ([[Phylicia Rashad]]), Apollo's widow, meets with Adonis and adopts him; informing him that he is [[Apollo Creed]]'s son (sired from an extramarital affair). Seventeen years later, in 2015, Adonis (using his biological mother's last name Johnson) is a wealthy young college graduate working at a securities firm at the Smith Boardley Financial Group. However, on weekends, he sneaks out to [[Tijuana]] to fight professional boxing matches against unheralded opponents and maintains an undefeated 15-0 record. Soon, Adonis resigns from his securities firm job to pursue his dream of becoming a boxer. Mary Anne vehemently opposes Adonis's plot of becoming a boxer, remembering how her husband was killed in the ring during a match against [[Ivan Drago]] [[Rocky IV|thirty years ago]] and how [[Rocky Balboa]] was forced into retirement after suffering [[cavum septum pellucidum|brain damage]]. Adonis finds it hard to get anyone in Los Angeles to train him due to his father's death in the ring, particularly after he suffers an embarrassing loss in a sparring match to light heavyweight contender Danny &quot;Stuntman&quot; Wheeler ([[Andre Ward]]). Undaunted, Adonis moves out of his mother's residence and travels to Philadelphia in hopes of seeking out his father's best friend and former rival, Rocky ([[Sylvester Stallone]]).

2824 And so when he got sick he was losing his strength because he had a muscular condition. He was having trouble getting around, having trouble carrying stuff. I started thinking about this idea of my dad's mortality. For me he was kind of like this mythical figure, my father, similar to what Rocky was for him. Going through it inspired me to make a film that told a story about his hero going through something similar to kind of motivate him and cheer him up. That's how I came up with the idea for this movie.&quot;&lt;ref&gt;{{cite web|title=Creed Director Ryan Coogler on His Chemistry With Michael B. Jordan|author=Eliana Dockterman|date=27 November 2015 |url=http://time.com/4128234/creed-director-ryan-coogler-michael-b-jordan/|access-date=November 27, 2015}}&lt;/ref&gt; Although Sylvester Stallone was initially reluctant to help out with the film,&lt;ref&gt;{{cite web|title='Creed': 15 Things to Know About the Latest 'Rocky' Installment|author=Justine Browning|website=[[Collider (website)|Collider]] |date=26 November 2015 |url=http://collider.com/creed-movie-rocky-spinoff-things-to-know/|access-date=November 27, 2015}}&lt;/ref&gt; he changed his mind upon meeting with Coogler and Jordan. In discussing Stallone's advice to him, Jordan said that he &quot;taught me how to throw punches and hit me in my chest a couple times.&quot;.&lt;ref&gt;{{cite web|title=Creed's Michael B. Jordan: Sylvester Stallone Gave Me Style Tips|author=Ashley Spencer|date=10 November 2015 |url=http://www.usmagazine.com/entertainment/news/creeds-michael-b-jordan-sylvester-stallone-gave-me-style-tips-20151011|access-date=November 27, 2015}}&lt;/ref&gt;

2825 Adonis is torn between trying to preserve his father's legacy and build his own. [[A.O. Scott]] of ''[[The New York Times]]'' wrote that, &quot;Adonis is a complex character with a complex fate. He is at once a rich kid and a street kid, the proud carrier of an illustrious heritage and an invisible man. His relationship with Rocky is complicated, too. The older fighter is a mentor and a father figure, to be sure, but he also needs someone to take care of him, especially when illness adds a melodramatic twist to the plot.&lt;ref&gt;{{cite news|title=Review: In 'Creed,' Rocky's Back, as a Mentor, Not a Fighter|author=A.O. Scott|work=The New York Times |date=24 November 2015 |url=https://www.nytimes.com/2015/11/25/movies/review-in-creed-rockys-back-as-a-mentor-not-a-fighter.html?&amp;_r=0|access-date=November 27, 2015}}&lt;/ref&gt; Adonis has been described as &quot;arrogant&quot;,.&lt;ref&gt;{{cite web|title=CREED IS WORTH BELIEVING IN|author=Richard Newby|url=http://www.audienceseverywhere.net/creed-is-worth-believing-in/|access-date=November 27, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=Sylvester Stallone and Michael B. Jordan triumph in 'Creed'|author=Gary Thompson|url=http://www.philly.com/philly/entertainment/movies/20151125_Sylvester_Stallone__Michael_B__Jordan_in__Creed_.html|access-date=November 27, 2015}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=Creed: The Passing of the Torch|author=Jeremy Dawson|url=http://afterovertime.com/boxing/creed-the-passing-of-the-torch/|access-date=November 27, 2015}}&lt;/ref&gt; Although Adonis' circumstances change after he is adopted by Mary Anne Creed, his late father's widow, he retains his fiery personality. Short-tempered and impulsive, but good-natured, it is Adonis' tenacity that convinces Rocky to train him.&lt;ref&gt;{{cite web|title=Yo Adrian: 'Creed' Is Really Good|author=Eric Eidelstein|website=[[Complex Networks]] |url=http://www.complex.com/pop-culture/2015/11/creed-movie-review|access-date=November 27, 2015}}&lt;/ref&gt; Michael O' Sullivan of ''[[The Washington Post]]'' analyzed that Adonis' &quot;struggles with his temper&quot; are &quot;a coping mechanism that helps him deal with the fear of not living up to the name Creed.&quot;&lt;ref&gt;{{cite news|title=Movie review: 'Creed' is a worthy successor to 'Rocky'|author=Michael O' Sullivan|newspaper=[[The Washington Post]] |url=https://www.washingtonpost.com/goingoutguide/movies/movie-review-creed-is-a-worthy-successor-to-rocky/2015/11/24/d7125c90-8fb7-11e5-baf4-bdf37355da0c_story.html|access-date=November 27, 2015}}&lt;/ref&gt;

2826 Bell is the brother of Mark Bell and [[Mike Bell (wrestler)|Mike Bell]], both of whom were featured in ''[[Bigger, Stronger, Faster*]]''.&lt;ref&gt;{{cite news|title=Q&amp;A / CHRIS BELL, DIRECTOR OF 'BIGGER, STRONGER, FASTER*: Steroid documentary hits close to home|last=Longino|first=Bob|date=13 June 2008|work=[[The Atlanta Journal-Constitution]]|page=3H}}&lt;/ref&gt; All three were overweight as children and strove, through competitive living and wrestling and, in the case of Bell, steroid use to emulate celebrities [[Hulk Hogan]], [[Sylvester Stallone]] and [[Arnold Schwarzenegger]].&lt;ref&gt;{{Cite news |last=Darling |first=Cary |date=17 August 2008 |title='Bigger Stronger Faster*': Powerful |volume=139 |page=9E |work=[[The Charlotte Observer]] |issue=230 |url=https://www.newspapers.com/clip/115289874/bigger-stronger-faster-movie-review/ |access-date=27 December 2022 |via=Newspapers.com}}&lt;/ref&gt;

2827 [[File:Sylvester Stallone 2012.jpg|thumb|right|150px|[[Sylvester Stallone]], Best Supporting Actor winner]]

2828 '''[[Sylvester Stallone]] - ''[[Creed (film)|Creed]]'' as [[Rocky Balboa]]'''

2829 |Guests=[[Sylvester Stallone]], [[John Stamos]]

2830 More was born in [[Hulme]], [[Manchester]] and began his career in entertainment as a singer after winning talent contests. He also made appearances on ''[[Blankety Blank]], [[Punchlines]]'' and ''[[The New Statesman (1987 TV series)|The New Statesman]]'' (1987) in which he portrayed Labour Party leader [[Neil Kinnock]]. His impersonations also included film stars; [[Michael Caine]], [[Sean Connery]], [[Jack Nicholson]], [[Arnold Schwarzenegger]] and [[Sylvester Stallone]] and he did impressions of [[Tony Blair]] and President [[George H. W. Bush]]. He died aged 81 in [[Salford Royal NHS Foundation Trust|Salford Royal Hospital]] from [[Leukaemia]].&lt;ref name=ManEveNews&gt;[http://www.manchestereveningnews.co.uk/news/greater-manchester-news/impressionist-johnny-more-star-who-10568664 impressionist Johnny More, star of Who Do You Do, dies aged 81] Retrieved 15 February 2016&lt;/ref&gt;

2831 * [[Sylvester Stallone]] - ''[[Creed (film)|Creed]]''

2832 ** ''[[Rocky III]]'' - [[Sylvester Stallone]]

2833 * '''[[Sylvester Stallone]] - ''[[Rocky III]]'''''

2834 ** [[Sylvester Stallone]] - ''[[First Blood]]''

2835 ** [[Sylvester Stallone]] - ''[[Rambo: First Blood Part II]]''

2836 ** ''[[Rocky IV]]'' - [[Sylvester Stallone]]

2837 ** [[Sylvester Stallone]] - ''[[Rocky IV]]''

2838 ** [[Sylvester Stallone]] - ''[[Over the Top (1987 film)|Over the Top]]''

2839 ** [[Sylvester Stallone]]&lt;!--voters were waiting for Rambo III--&gt;

2840 ** [[Sylvester Stallone]] - ''[[Creed (film)|Creed]]''

2841 ** [[Sylvester Stallone]] - ''[[Creed (film)|Creed]]''

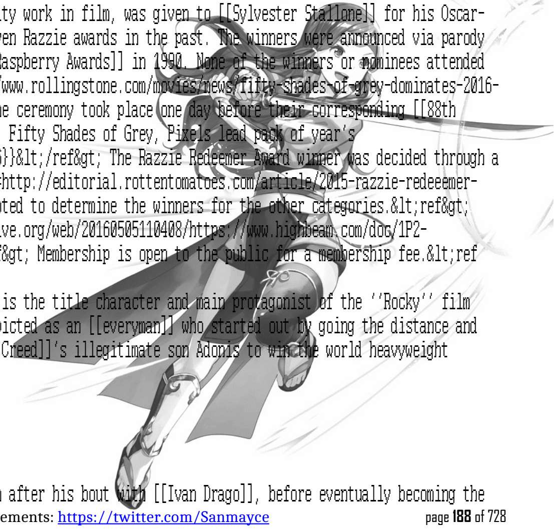
2842 '''Gaetano Marco &quot;Guy&quot; Nardulli''' (born May 31, 1974 in [[Norridge, Illinois]]) is an American actor and producer who is most associated for his character role as Johnny The Zip on Sylvester Stallone's show TULSA KING (2023-2024) ''The Horror Vault Vol.1'' (2008), which was released as a compilation of nine horror short stories&lt;ref&gt;{{cite web|last1=Nardulli|first1=Guy|title=The Horror Vault Vol. 1|url=https://www.imdb.com/title/tt0369424/|website=IMDb}}&lt;/ref&gt; contains the 2005 movie short thriller ''Alone'' in which Nardulli plays Detective Wiley. Guy is well known for his &quot;strong-arm&quot; portrayals as detective and law enforcement characters, military, mobster and criminal roles in movies and television. His most recent television work is related with the 2023 television series ''[[Tulsa King]]'' Where he appears as a series regular in season 2 (2024) ''[[The Last Ship (TV series)|The Last Ship]]'' in two episodes, &quot;Long Day's Journey&quot; and &quot;Alone and Unafraid&quot;. Guy also appears in a recurring role on ''[[Criminal Minds]]'' as Detective Walker. He was nominated for Best Supporting Actor for the 2013 [[Utah Film Awards]] for his role as Antonio Sorrento for ''[[Proper Manors]]''. Nardulli also received a second nomination as part of the Best Ensemble Nomination, also for Proper Manors.

2843 ** [[Sylvester Stallone]] - ''[[Creed (film)|Creed]]'' as [[Rocky Balboa]]

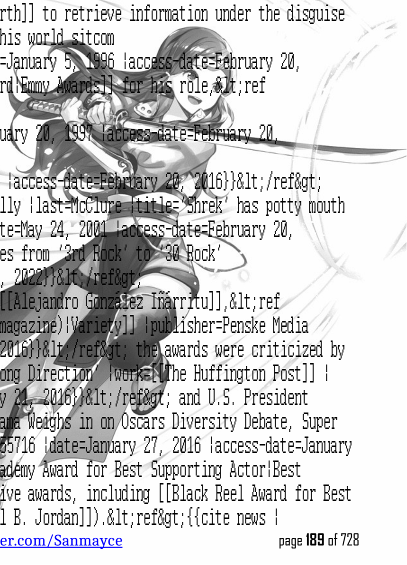
2844 * [[Sylvester Stallone]] - ''[[Creed (film)|Creed]]'' as [[Rocky Balboa]]

2845 || Director: [[Patrick Hughes (director)|Patrick Hughes]] &lt;br&gt; Cast: [[Sylvester Stallone]], [[Jason Statham]], [[Antonio Banderas]], [[Jet Li]], [[Wesley Snipes]], [[Dolph Lundgren]], [[Kelsey Grammer]], [[Randy Couture]], [[Terry Crews]], [[Kellan Lutz]], [[Ronda Rousey]], [[Glen Powell]], [[Victor Ortiz]], [[Robert Davi]], [[Mel Gibson]], [[Harrison Ford]], [[Arnold Schwarzenegger]]

```
In addition, Koenig modified cars owned by celebrities. These include actor [[Sylvester Stallone]], who had a silver [[Mercedes-Benz W126|Mercedes-Benz 560SEC]] and [[Formula One]] driver [[Gerhard Berger]], who had a 650ps twin-turbo Testarossa.{{sfn|Tücherer|1985|pp=52-54}}{{sfn|Tücherer|1988|pp=42-45}}
* [[Sylvester Stallone]] and [[Carl Weathers]] with Best Motion Picture - Drama
* [[Sylvester Stallone]]
| based_on       = {{Based on|[[List of Rocky characters|Characters]]|Sylvester Stallone}}
* Sylvester Stallone
* Sylvester Stallone
'''''Creed II''''' is a 2018 American [[Sports film|sports]] [[Drama (film and television)|drama film]] directed by [[Steven Caple Jr.]] from a screenplay by [[Juel Taylor]] and [[Sylvester Stallone]]. It is the sequel to ''[[Creed (film)|Creed]]'' (2015) and the eighth installment in the [[Rocky (franchise)|''Rocky'' film series]]. It stars [[Michael B. Jordan]], Stallone, [[Tessa Thompson]], [[Wood Harris]], [[Phylicia Rashad]], [[Florian Munteanu]], and [[Dolph Lundgren]]. In the film, under the continued tutelage of [[Rocky Balboa]] (Stallone), [[Adonis Creed]] (Jordan) faces off against Viktor Drago (Munteanu), the son of [[Ivan Drago]] (Lundgren), who became responsible for the death of Adonis' father [[Apollo Creed]] in ''[[Rocky IV]]'' (1985).
* [[Sylvester Stallone]] as [[Rocky Balboa|Robert &quot;Rocky&quot; Balboa]]: A two-time world heavyweight champion and Apollo's best friend and former rival who becomes Adonis' avuncular trainer and mentor. He owns and operates an Italian restaurant in Philadelphia named after his deceased wife, Adrian.
On January 5, 2016, [[Sylvester Stallone]] and [[Metro-Goldwyn-Mayer Pictures]] CEO [[Gary Barber]] confirmed to ''[[Variety (magazine)|Variety]]'' that a sequel to ''Creed'' was in development.&lt;ref name=&quot;InDevelopment&quot; /&gt; That month, Stallone posed the possibility of [[Milo Ventimiglia]] reprising his role as Rocky's son Robert Balboa from ''[[Rocky Balboa (film)|Rocky Balboa]]''. Ventimiglia had revealed during the development of ''Creed'' that he was open to returning to the franchise, stating, &quot;I'll tell you what, if they invited me, I'd love to be there. If they didn't, I wouldn't be offended.&quot;&lt;ref&gt;{{cite news|last1=Topel|first1=Fred|title=Exclusive: Milo Ventimiglia Ready for Creed Comeback|url=http://www.craveonline.com/site/545137-exclusive-milo-ventimiglia-ready-for-creed-comeback|access-date=September 28, 2017|publisher=Crave|date=August 3, 2013|archive-url=https://web.archive.org/web/20170928152823/http://www.craveonline.com/site/545137-exclusive-milo-ventimiglia-ready-for-creed-comeback|archive-date=September 28, 2017|url-status=dead}}&lt;/ref&gt; It was revealed in April 2018 that he had been cast.&lt;ref name=&quot;phillychitchat.com&quot;&gt;{{Cite web |url=http://www.phillychitchat.com/2018/04/ventimiglia.html |title=Creed 2: Milo Ventimiglia - This Is Us - Returns and More from the Set - Philly Chit Chat |access-date=September 22, 2018 |archive-url=https://web.archive.org/web/20180922064106/http://www.phillychitchat.com/2018/04/ventimiglia.html |archive-date=September 22, 2018 |url-status=dead }}&lt;/ref&gt; On January 11, 2016, Barber revealed that [[Ryan Coogler]] would not be returning due to scheduling conflicts with ''[[Black Panther (film)|Black Panther]]'', though he would return as executive producer.&lt;ref name=&quot;CooglerLeft&quot; /&gt; [[Michael B. Jordan]] was paid between $3 and $4 million,&lt;ref&gt;{{Cite web |last=Lang |first=Brent |date=May 8, 2018 |title=Leonardo DiCaprio, Jennifer Lawrence and Other Star Salaries Revealed |url=https://variety.com/2018/film/news/celebrity-salaries-daniel-craig-jennifer-lawrence-leonardo-dicaprio-1202801717/ |url-status=live |archive-url=https://web.archive.org/web/20180618152314/https://variety.com/2018/film/news/celebrity-salaries-daniel-craig-jennifer-lawrence-leonardo-dicaprio-1202801717/ |archive-date=June 18, 2018 |access-date=February 26, 2023 |website=[[Variety (magazine)|Variety]]}}&lt;/ref&gt; and his schedule was delayed by starring in ''Black Panther''.&lt;ref name=&quot;CooglerLeft&quot; /&gt; In July 2017, Stallone confirmed that he had completed the script, and that [[Ivan Drago]] would be featured.&lt;ref name=&quot;ScripCompleted&quot; /&gt; In October 2017, it was announced that Stallone would direct and produce the film.&lt;ref&gt;{{Cite news|url=https://www.vanityfair.com/hollywood/2017/10/sylvester-stallone-creed-2|title=He's Back: Sylvester Stallone Will Direct Creed 2|last=Desta|first=Yohana|work=HWD|access-date=March 19, 2018|language=en}}&lt;/ref&gt; However, in December 2017, it was reported that [[Steven Caple Jr.]] would instead direct the film with [[Tessa Thompson]] confirmed to reprise her role of Bianca, Creed's love interest.&lt;ref&gt;{{Cite news|url=https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/|title=Steven Caple Jr. to Direct 'Creed 2' Starring Michael B. Jordan and Sylvester Stallone (Exclusive)|last=Kroll|first=Justin|date=December 11, 2017|work=[[Variety (magazine)|Variety]]|access-date=March 19, 2018|language=en-US}}&lt;/ref&gt; In January 2018, Romanian amateur boxer Florian Munteanu was cast as Drago's son, and [[Dolph Lundgren]] to reprise his role of Drago.&lt;ref name=&quot;MunteanuCast&quot;&gt;{{cite news|last1=Sneider|first1=Jeff|title=&quot;Creed 2&quot;: Florian &quot;Big Nasty&quot; Munteanu to Play Dolph Lundgren's Son and Michael B. Jordan's Opponent (Exclusive)|url=http://www.tracking-board.com/creed-2-florian-big-nasty-munteanu-to-play-dolph-lundgrens-son-and-michael-b-jordans-opponent-exclusive/|access-date=January 17, 2018|work=The Tracking Board|date=January 15, 2018}}&lt;/ref&gt; In March 2018, [[Russell Hornsby]] joined the cast and [[Phylicia Rashad]], [[Wood Harris]], and [[Andre Ward]] were confirmed to reprise their roles from the prior film.&lt;ref&gt;{{Cite news|url=https://deadline.com/2018/04/russell-hornsby-creed-2-melvin-gregg-high-flying-bird-1202356880/|title=Russell Hornsby Joins 'Creed II'; Melvin Gregg Cast In 'High Flying Bird'|last=N'Duka|first=Amanda|date=April 2, 2018|work=[[Deadline Hollywood]]|access-date=April 3, 2018|language=en-US}}&lt;/ref&gt;
In response to the suggestion that [[Deontay Wilder]] could play the son of Clubber Lang in a potential ''Creed III'', both Sylvester Stallone and Michael B. Jordan expressed interest in such a character being featured in the plot of the next installment.&lt;ref&gt;{{cite web |title=Michael B. Jordan Clubber Lang's Son in 'Creed 3? ... I Like It! |url=https://www.tmz.com/2018/12/26/michael-b-jordan-creed-3-deontay-wilder-clubber-lang/ |website=TMZ Sports|date=December 26, 2018 }}&lt;/ref&gt;&lt;ref&gt;{{cite web |title=Sylvester Stallone says Deontay Wilder could play Clubber Lang's son in Creed III |url=https://www.cbssports.com/boxing/news/sylvester-stallone-says-deontay-wilder-could-play-clubber-langs-son-in-creed-iii/ |website=CBS Sports|date=December 13, 2018 }}&lt;/ref&gt; In February 2020, Zach Baylin was announced as the sequel's writer.&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/heat-vision/creed-3-taps-king-richard-writer-zach-baylin-1281094 |title='Creed 3' Taps 'King Richard' Writer (Exclusive) |last=Galuppo |first=Mia |website=[[The Hollywood Reporter]] |date=February 25, 2020 |access-date=February 25, 2020}}&lt;/ref&gt; In October 2020, it was reported that Michael B. Jordan would reprise his role of Adonis Creed, and have his directorial debut.&lt;ref&gt;{{Cite web|last=Fleming|first=Mike Jr.|date=2020-10-24|title=Don't Hold Your Breath Waiting To See 007 Film 'No Time To Die' On Streamer As Cursory Talks Died Quickly|url=https://deadline.com/2020/10/james-bond-no-time-to-die-streamer-talks-died-quickly-1234602960/|access-date=2020-10-24|website=Deadline|language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|last=Gemmill|first=Allie|date=2020-10-24|title=Michael B. Jordan to Make Directorial Debut With 'Creed 3' in Addition to Starring|url=https://collider.com/creed-3-michael-b-jordan-director/|access-date=2020-10-24|website=Collider|language=en-US}}&lt;/ref&gt; Jordan was confirmed as the director of ''Creed III'' in March 2021, with a targeted release date of March 3, 2023,&lt;ref&gt;{{cite web|url=https://www.comingsoon.net/movies/news/1232799-creed-iii-delayed-to-2023-new-release-date-set|title=Creed III Delayed to 2023, New Release Date Set|work=ComingSoon.net|last=Treese|first=Tyler|date=July 28, 2022|access-date=July 28, 2022}}&lt;/ref&gt; and Stallone confirmed in April that he would not appear as Rocky Balboa.&lt;ref&gt;{{cite web |url=https://www.joblo.com/movie-news/creed-3-sylvester-stallone-rocky |title='Creed 3': Sylvester Stallone's Rocky Sitting This Round Out |last=Bumbray |first=Chris |website=[[JoBlo.com]] |date=April 5, 2021 |access-date=April 5, 2021 |archive-url=https://web.archive.org/web/20210405194513/https://www.joblo.com/movie-news/creed-3-sylvester-stallone-rocky |archive-date=April 5, 2021 |url-status=live}}&lt;/ref&gt;
As of July 2019, the ''Rocky'' franchise as a whole was announced to continue with another mentor-student film like ''Creed'' and set after ''Creed II'', in which Rocky Balboa befriends a young fighter who is also an illegal immigrant. Stallone stated: &quot;Rocky meets a young, angry person who got stuck in this country when he comes to see his sister. He takes him into his life, and unbelievable adventures begin, and they wind up south of the border. It's very, very timely.&quot; In addition he announced the development of a ''Rocky'' prequel television series.&lt;ref&gt;{{cite web|url=https://variety.com/2019/film/features/sylvester-stallone-rocky-ownership-stake-1203275639/ |title=Sylvester Stallone Feels Robbed of an Ownership Stake in 'Rocky': 'I Was Furious' |work=[[Variety (magazine)|Variety]] |date=July 23, 2019 |access-date=July 23, 2019}}&lt;/ref&gt;
&lt;ref name=&quot;InDevelopment&quot;&gt;{{cite magazine|last1=Setoodeh|first1=Ramin|title=Sylvester Stallone Says 'Creed' Sequel Could Reunite Rocky and Apollo (EXCLUSIVE)|url=https://variety.com/2016/film/news/creed-sequel-sylvester-stallone-1201672298/|access-date=January 6, 2016|magazine=[[Variety (magazine)|Variety]]|date=January 5, 2016}}&lt;/ref&gt;
{{Sylvester Stallone}}
[[Category:Films with screenplays by Sylvester Stallone]]
:* '''[[Sylvester Stallone]] – ''[[Creed (film)|Creed]]'''''
''[[Fifty Shades of Grey (film)|Fifty Shades of Grey]]'' received the most awards, winning [[Golden Raspberry Award for Worst Picture|Worst Picture]], [[Golden Raspberry Award for Worst Actor|Worst Actor]], [[Golden Raspberry Award for Worst Actress|Worst Actress]], [[Golden Raspberry Award for Worst Screen Combo|Worst Screen Combo]] and [[Golden Raspberry Award for Worst Screenplay|Worst Screenplay]]. ''[[Fantastic Four (2015 film)|Fantastic Four]]'' tied for Worst Picture, also receiving [[Golden Raspberry Award for Worst Director|Worst Director]] and [[Golden Raspberry Award for Worst Prequel, Remake, Rip-off or Sequel|Worst Remake, Rip-off or Sequel]].&lt;ref name=&quot;CNN&quot; /&gt; [[Eddie Redmayne]] received [[Golden Raspberry Award for Worst Supporting Actor|Worst Supporting Actor]] for ''[[Jupiter Ascending]]'', while [[Kaley Cuoco]] received [[Golden Raspberry Award for Worst Supporting Actress|Worst Supporting Actress]] for ''[[Alvin and the Chipmunks: The Road Chip]]'' and ''[[The Wedding Ringer]]''. The [[Razzie Redeemer Award]], given to former Razzie winners and nominees for quality work in film, was given to [[Sylvester Stallone]] for his Oscar-nominated role in ''[[Creed (film)|Creed]]'' after [[List of awards and nominations received by Sylvester Stallone|receiving]] seven Razzie awards in the past. The winners were announced via parody sketches; one of them was performed by an impersonation of [[Donald Trump]], who won Worst Supporting Actor at the [[11th Golden Raspberry Awards]] in 1990. None of the winners or nominees attended the ceremony.&lt;ref&gt;{{cite magazine|last1=Kreps|first1=Daniel|title='Fifty Shades of Grey' Dominates 2016 Razzies|url=https://www.rollingstone.com/movies/news/fifty-shades-of-grey-dominates-2016-razzies-20160228|magazine=Rolling Stone|access-date=11 March 2016}}&lt;/ref&gt; Per tradition, the nominations announcement and the ceremony took place one day before their corresponding [[88th Academy Awards]] ceremony.&lt;ref name=&quot;ew&quot;&gt;{{cite web|last1=Rosen|first1=Christopher|title=Razzies nominations 2016: Fifty Shades of Grey, Pixels lead pack of year's worst|url=http://www.ew.com/article/2016/01/13/razzies-nominations-2016-list|website=Entertainment Weekly|access-date=1 March 2016}}&lt;/ref&gt; The Razzie Redeemer Award winner was decided through a public vote on [[Rotten Tomatoes]],&lt;ref&gt;{{cite web|title=Pick the &quot;Winner&quot; for the 2015 Razzie Redeemer Award|url=http://editorial.rottentomatoes.com/article/2015-razzie-redeeemer-award/|website=Rotten Tomatoes|access-date=11 March 2016}}&lt;/ref&gt; while the 943 members of the Golden Raspberry Foundation voted to determine the winners for the other categories.&lt;ref&gt;{{cite web|title='Fifty Shades' Dominates Razzies|url=https://www.highbeam.com/doc/1P2-39359134.html|archive-url=https://web.archive.org/web/20160505110408/https://www.highbeam.com/doc/1P2-39359134.html|url-status=dead|archive-date=5 May 2016|website=[[Telegraph Herald]]|access-date=3 March 2016|url-access= }}&lt;/ref&gt; Membership is open to the public for a membership fee.&lt;ref name=&quot;AFP&quot; /&gt;
'''Robert &quot;Rocky&quot; Balboa''' (also known by his ring name '''The Italian Stallion''', played by [[Sylvester Stallone]]), is the title character and main protagonist of the ''Rocky'' film series. The character was created by Sylvester Stallone, who has also portrayed him in all eight films in the franchise. He is depicted as an [[everyman]] who started out by going the distance and overcoming obstacles that had occurred in his life and career as a professional boxer. In the ''Creed'' films, he trains [[Apollo Creed]]'s illegitimate son Adonis to win the world heavyweight championship.
[[File:Sylvester Stallone Cannes 2019.jpg|thumb|right|150px|[[Sylvester Stallone]], Razzie Redeemer Award winner]]
| creator = [[Sylvester Stallone]]
* '''[[Sylvester Stallone]] – From all-time Razzie champ to 2015 award contender for ''[[Creed (film)|Creed]]'''''
| creator = [[Sylvester Stallone]]
'''Tony &quot;Duke&quot; Evers''' was initially the manager/trainer to the world champion [[Apollo Creed]] until his in-ring death after his bout with [[Ivan Drago]], before eventually becoming the
```



```
manager to Apollo's friend [[Rocky Balboa]]. He was portrayed by [[Tony Burton]] and is one of only four characters (along with [[Rocky Balboa|Rocky]], Paulie Pennino, and [[Stu Nahan]]) to have been featured in all the original six ''Rocky'' films.&lt;ref&gt;{{cite book|url=https://books.google.com/books?id=nVIZBQAAQBAJ&amp;q=TonyDukeEverstonyburton&amp;pg=PA189 |title=The Ultimate Stallone Reader: Sylvester Stallone as Star, Icon, Auteur |page=189 |date=2014-04-14 |isbn=9780231169806 |access-date=2016-02-18|last1=Holmlund |first1=Chris }}&lt;/ref&gt;
| creator     = [[Sylvester Stallone]]
In real-life, [[Brigitte Nielsen]] was engaged to [[Sylvester Stallone]] (who plays Rocky), during production of ''Rocky IV'', and married shortly after the release. Both of them once again acted together in ''[[Cobra (1986 film)|Cobra]]'' (1986), before they divorced in 1987.
| creator            = [[Sylvester Stallone]]
| creator            = [[Sylvester Stallone]]
| creator            = [[Sylvester Stallone]]
* [[Dustin Hoffman]] / [[Sylvester Stallone]] (1977)
** [[Sylvester Stallone]]&amp;nbsp;- ''[[Creed (film)|Creed]]''
|[[Sylvester Stallone]]
|[[File:Sylvester Stallone 2012.jpg|left|100x100px]]
| style=&quot;text-align: left;&quot; | [[:Sylvester Stallone]]
| style=&quot;text-align: left;&quot; | [[:Sylvester Stallone]]
* [[Sylvester Stallone]]
'''''The Suicide Squad''''' is a 2021 American [[superhero film]] based on the [[DC Comics]] team [[Suicide Squad]]. Produced by [[DC Films]], [[Atlas Entertainment]], and [[the Safran Company]], and distributed by [[Warner Bros. Pictures]], it is a [[standalone sequel]] to ''[[Suicide Squad (2016 film)|Suicide Squad]]'' (2016) and the 10th film in the [[DC Extended Universe]] (DCEU). The film was written and directed by [[James Gunn]] and stars an [[ensemble cast]] including &lt;!--Order based on title credits per talk page consensus--&gt;[[Margot Robbie]], [[Idris Elba]], [[John Cena]], [[Joel Kinnaman]], [[Sylvester Stallone]], [[Viola Davis]], [[David Dastmalchian]], [[Daniela Melchior]], [[Michael Rooker]], [[Jai Courtney]], [[Peter Capaldi]], [[Alice Braga]], and [[Pete Davidson]].&lt;!--Order based on title credits per talk page consensus--&gt; In the film, several convicts join a task force known as the &quot;Suicide Squad&quot; in exchange for lighter sentences. They are sent to the [[South America]]n island nation of [[Corto Maltese (DC Comics)|Corto Maltese]] to destroy all traces of the giant alien starfish [[Starro the Conqueror]] before it falls into the local government's control.
| image3  = Sylvester Stallone by Gage Skidmore.jpg
| alt3    = Sylvester Stalone at the 2010 Comic Con in San Diego.
| footer  = The film's second Suicide Squad team includes [[Idris Elba]] as [[Robert DuBois (DC Extended Universe)|Bloodsport]], [[John Cena]] as [[Peacemaker (DC Extended Universe character)|Peacemaker]], [[Sylvester Stallone]] as the voice of [[Nanaue (DC Extended Universe)|King Shark]], and [[Daniela Melchior]] as [[Cleo Cazo|Ratcatcher 2]].}}
* [[Sylvester Stallone]] as the voice of [[Nanaue (DC Extended Universe)|Nanaue / King Shark]]:&lt;br /&gt;A man-eating shark-human hybrid,&lt;ref name=&quot;Stallone&quot; /&gt;&lt;ref name=&quot;VergeTrailer&quot; /&gt; created with visual effects.&lt;ref name=&quot;GunnKSRealPerson&quot; /&gt; [[Steve Agee]], a longtime friend of Gunn's, developed King Shark's portrayal on set. He was chosen partly because he is {{convert|6|ft|7|in|m}}, and wore a wireframe version of a shark head as well as a body suit to further approximate the character's size and shape.&lt;ref name=&quot;AgeeTall&quot; /&gt; Gunn wanted the underlying story for King Shark to explore his loneliness as someone who wants to be part of the world and connect with other people, but is not able to due to his appearance. This is seen in moments where he pretends to read or looks longingly out a window at a couple who are together.&lt;ref name=&quot;GunnScriptApart&quot; /&gt;
Gunn wrote King Shark with [[Sylvester Stallone]] in mind,&lt;ref name=&quot;DoGGunnApr2021KingShark&quot; /&gt;&lt;ref name=&quot;StalloneInMind&quot; /&gt; but tested three other voice actors before asking Stallone to join the film.&lt;ref name=&quot;GunnKSCB&quot; /&gt; Stallone agreed to voice the character due to his experience working with Gunn on ''[[Guardians of the Galaxy Vol. 2]]'' (2017).&lt;ref name=&quot;DoGGunnApr2021KingShark&quot; /&gt;&lt;ref name=&quot;StalloneInMind&quot; /&gt; Gunn was looking to cast Bautista as Peacemaker,&lt;ref name=&quot;ColliderCharacters&quot; /&gt; but the actor had a scheduling conflict with Zack Snyder's ''[[Army of the Dead]]'' (2021).&lt;ref name=&quot;BautistaScheduling&quot; /&gt;&lt;ref name=&quot;BautistaSnyder&quot; /&gt; [[John Cena]] entered talks for a role in the film that was believed to be Peacemaker in late April,&lt;ref name=&quot;BautistaScheduling&quot; /&gt;&lt;ref name=&quot;Cena&quot; /&gt; as Gunn had wanted to work with Cena since seeing his performance in ''[[Trainwreck (film)|Trainwreck]]'' (2015).&lt;ref name=&quot;GunnCenaCasting&quot; /&gt; Cena made numerous unsuccessful attempts to join the DCEU prior to his casting in this film.&lt;ref name=&quot;DoGCenaApr2021&quot; /&gt; [[David Dastmalchian]] and [[Daniela Melchior]] were respectively cast as Polka-Dot Man and Ratcatcher 2 at the end of April.&lt;ref name=&quot;Dastmalchian&quot; /&gt;&lt;ref name=&quot;Melchior&quot; /&gt; Gunn said the role of Polka-Dot Man was &quot;tailor made&quot; for Dastmalchian, who had been friends with Gunn for a long time.&lt;ref name=&quot;GunnColliderAug2021&quot; /&gt; In contrast, Melchior was cast from a group of 200 actors.&lt;ref name=&quot;THRGunnGrounded&quot; /&gt; She did a chemistry test with live rats as part of her audition process because her character controls rats in the film and had to interact with them on set.&lt;ref name=&quot;EsquireJul2021&quot; /&gt; Gunn's frequent collaborator [[Michael Rooker]] was in talks to join the cast in May,&lt;ref name=&quot;Rooker&quot; /&gt; while [[Storm Reid]] was cast as the daughter of Elba's character in July.&lt;ref name=&quot;Reid&quot; /&gt; [[Flula Borg]], [[Nathan Fillion]], and [[Steve Agee]] joined the film in August,&lt;ref name=&quot;Borg&quot; /&gt;&lt;ref name=&quot;Fillion&quot; /&gt;&lt;ref name=&quot;Agee&quot; /&gt; with Agee initially reported to be portraying King Shark.&lt;ref name=&quot;Agee&quot; /&gt; Also in August, [[Taika Waititi]] entered negotiations for a role.&lt;ref name=&quot;Waititi&quot; /&gt; [[Peter Capaldi]] joined the cast in early September, when [[Pete Davidson]] was in talks to make a [[cameo appearance]] during a break from his work on ''[[Saturday Night Live]]''.&lt;ref name=&quot;CapaldiDavidson&quot; /&gt; Davidson agreed to join the film because his character is called &quot;[[100 (DC Comics)#1,000 Operatives|Dick Hertz]]&quot;, which Davidson found to be funny.&lt;ref name=&quot;BlackguardName&quot; /&gt;
| data-sort-value=&quot;Stallone, Sylvester&quot; | [[Sylvester Stallone]]
&lt;ref name=&quot;Stallone&quot;&gt;{{Cite web |last=Haring |first=Bruce |date=November 14, 2020 |title='The Suicide Squad' Adds Sylvester Stallone To Its Lineup, Confirmed By James Gunn On Instagram |url=https://deadline.com/2020/11/the-suicide-squad-adds-sylvester-stallone-james-gunn-confims-on-instagram-1234615905/ |url-status=live |archive-url=https://web.archive.org/web/20201114232924/https://deadline.com/2020/11/the-suicide-squad-adds-sylvester-stallone-james-gunn-confims-on-instagram-1234615905/ |archive-date=November 14, 2020 |access-date=November 14, 2020 |website=[[Deadline Hollywood]]}}&lt;/ref&gt;
&lt;ref name=&quot;GunnKSCB&quot;&gt;{{Cite web |last=O'Connell |first=Sean |date=July 15, 2021 |title=Why King Shark Was A 'Very, Very Difficult' Suicide Squad Character To Create, Until Sylvester Stallone Helped |url=https://www.cinemablend.com/news/2570471/why-king-shark-was-very-difficult-suicide-squad-character-to-create-sylvester-stallone-dc |url-status=live |archive-url=https://web.archive.org/web/20210716020343/https://www.cinemablend.com/news/2570471/why-king-shark-was-very-difficult-suicide-squad-character-to-create-sylvester-stallone-dc |archive-date=July 16, 2021 |access-date=July 15, 2021 |website=[[CinemaBlend]]}}&lt;/ref&gt;
&lt;ref name=&quot;StalloneReprisal&quot;&gt;{{Cite web |last=Perry |first=Spencer |date=November 10, 2022 |title=Sylvester Stallone Confirms King Shark Return (Exclusive) |url=https://comicbook.com/dc/news/suicide-squad-sylvester-stallone-king-shark-return/ |url-status=live |archive-url=https://web.archive.org/web/20221124074313/https://comicbook.com/dc/news/suicide-squad-sylvester-stallone-king-shark-return/ |archive-date=November 24, 2022 |access-date=December 7, 2022 |website=[[ComicBook.com]]}}&lt;/ref&gt;
| '''[[Sylvester Stallone]]'''&lt;ref&gt;[http://www.comingsoon.net/movies/news/661257-razzie-awards-2016-winners-announced#/slide/1 Razzie Awards 2016 Winners Announced] {{Webarchive|url=https://web.archive.org/web/20160311102520/http://www.comingsoon.net/movies/news/661257-razzie-awards-2016-winners-announced#/slide/1 |date=March 11, 2016 }} Comingsoon.net, Retrieved February 28, 2016&lt;/ref&gt;
The 1990s saw Lithgow continue to appear in various Hollywood films, namely ''[[Ricochet (1991 film)|Ricochet]]'' (1991) opposite [[Denzel Washington]], ''[[Raising Cain]]'' (1992), ''[[Cliffhanger (film)|Cliffhanger]]'' (1993) starring [[Sylvester Stallone]] and ''[[The Pelican Brief (film)|The Pelican Brief]]'' (1993), where he was reunited with Washington.&lt;ref name=&quot;Lithgow_hollywood&quot;/&gt;&lt;ref&gt;{{cite news |title=Press of events readies Lithgow for editor role |url=https://news.google.com/newspapers?nid=336&amp;dat=19930730&amp;id=Cj9TAAAAIBAJ&amp;pg=6054,6520033&amp;hl=sv |newspaper=The Deseret News |date=July 30, 1993 |access-date=February 20, 2016}}&lt;/ref&gt; He was cast in a main role in the 1996 television sitcom ''[[3rd Rock from the Sun]]'', where he played a high-ranking commander of an alien unit of four who have been sent to [[Earth]] to retrieve information under the disguise as a university professor.&lt;ref name=&quot;3rdrockfromthesun&quot;&gt;{{cite news |first=John |last=Crook |title=Lithgow, Curtin sparkle in out-of-this world sitcom |url=https://news.google.com/newspapers?nid=2457&amp;dat=19960105&amp;id=Z5xhAAAAIBAJ&amp;pg=4718,1594353&amp;hl=sv |newspaper=Bangor Daily News |date=January 5, 1996 |access-date=February 20, 2016}}&lt;/ref&gt; The show spanned over 100 episodes, during which Lithgow won one [[Golden Globe Award|Golden Globe]] and three [[Primetime Emmy Award|Emmy Awards]] for his role,&lt;ref name=&quot;emmys&quot;/&gt;&lt;ref&gt;{{cite news |title='English Patient,' 'Evita,' Madonna, Tom Cruise and 'X-Files' big Golden Globe winners |url=http://chronicle.augusta.com/stories/1997/01/20/ent_202824.shtml#.Vrqy38fiNQI |agency=Associated Press (AP) |work=The Augusta Chronicle |date=January 20, 1997 |access-date=February 20, 2016}}&lt;/ref&gt; before ending in 2001.&lt;ref&gt;{{cite news |first=Lynn |last=Elber |title='3rd Rock from the Sun' ends six-year run |url=http://amarillo.com/stories/2001/05/22/ent_rockfrom.shtml#.Vrqrc8fiNQI |agency=Associated Press (AP) |work=Amarillo Globe-News |date=May 22, 2001 |access-date=February 20, 2016}}&lt;/ref&gt; That same year, he became the character of [[Lord Farquaad]] in the animated fantasy-comedy film ''[[Shrek]]'' (2001).&lt;ref&gt;{{cite news |first=Holly |last=McClure |title='Shrek' has potty mouth |url=https://news.google.com/newspapers?nid=1298&amp;dat=20010524&amp;id=duwyAAAAIBAJ&amp;pg=4068,6491624&amp;hl=sv |newspaper=The Free Lance-Star |date=May 24, 2001 |access-date=February 20, 2016}}&lt;/ref&gt;&lt;ref name=&quot;Shrekthethird&quot;&gt;{{cite news |first=Mark |last=Dawidziak |author-link=Mark Dawidziak |title=John Lithgow goes from '3rd Rock' to '30 Rock' |url=https://www.cleveland.com/tv/2009/03/john_lithgow_goes_from_3rd_roc.html |work=Cleveland Plain Dealer |date=March 4, 2009 |access-date=January 25, 2022}}&lt;/ref&gt;
After not a single black actor or filmmaker was nominated in any of the major categories and with Hispanic filmmakers again only being represented by [[Alejandro González Iñárritu]],&lt;ref name=&quot;Variety 2016-01-14&quot;&gt;{{cite news |author=Tim Gray |title=Academy Nominates All White Actors for Second Year in Row |work=[[Variety (magazine)|Variety]] |publisher=Penske Media Corporation |url=https://variety.com/2016/biz/news/oscar-nominations-2016-diversity-white-1201674903/ |date=January 14, 2016 |access-date=January 18, 2016}}&lt;/ref&gt; the awards were criticized by celebrities including [[George Clooney]]&lt;ref&gt;{{cite news |author=Julia Brucculieri |title=George Clooney Thinks The Oscars Are 'Moving In The Wrong Direction' |work=[[The Huffington Post]] |url=http://www.huffingtonpost.com/entry/george-clooney-oscars-diversity-problem_us_569e8054e4b00f3e98630c5c |date=January 19, 2016 |access-date=January 21, 2016}}&lt;/ref&gt; and U.S. President [[Barack Obama]], who asked &quot;Are we making sure that everybody is getting a fair shot?&quot;&lt;ref&gt;{{cite news |author=Avery Miller |title=Obama Weighs in on Oscars Diversity Debate, Super Bowl |publisher=[[ABC News (United States)|ABC News]] |url=https://abcnews.go.com/Politics/obama-weighs-oscar-diversity-debate-super-bowl/story?id=36555716 |date=January 27, 2016 |access-date=January 28, 2016}}&lt;/ref&gt; Coogler's widely praised film ''[[Creed (film)|Creed]]'' earned just a single nomination for [[Sylvester Stallone]] in the [[Academy Award for Best Supporting Actor|Best Supporting Actor]] category, while it was the great winner at the recent [[Black Reel Awards of 2016|Black Reel Awards]], where it earned a total of five awards, including [[Black Reel Award for Best Film|Best Film]], [[Black Reel Award for Best Director|Best Director]] (for Coogler), and [[Black Reel Award for Best Actor|Best Actor]] (for [[Michael B. Jordan]]).&lt;ref&gt;{{cite news |
```



title=&quot;Creed&quot; Takes Top Honors at 16th Annual Black Reel Awards ¦agency=[[ABC News Radio]] ¦work=[[WBT (radio station)¦WBT]] ¦url=http://wbt.com/creed-takes-top-honors-at-16th-annual-black-reel-awards/ ¦date=February 19, 2016 ¦access-date=February 28, 2016}}&lt;/ref&gt;

2893 A number of celebrities including [[Will Smith]], Best Original Song nominee [[Antony Hegarty]],&lt;ref name=&quot;AnohniNOM&quot;&gt;{{cite web ¦author=Tim Jonze ¦title=Oscars 2016: Anohni boycotts 'degrading' ceremony ¦work=[[The Guardian]] ¦url=https://www.theguardian.com/music/2016/feb/26/anohni-boycotts-oscars-ceremony-transgender-diversity ¦date=February 26, 2016 ¦access-date=February 27, 2016}}&lt;/ref&gt; [[Jussie Smollett]],&lt;ref&gt;{{cite web ¦author=Lior Zaltzman ¦title=Jewish 'Empire' Star Ditches Oscars for #JusticeForFlint ¦work=[[The Forward]] ¦url=http://forward.com/the-assimilator/334372/jewish-empire-star-ditches-oscars-for-justiceforflint/ ¦date=January 28, 2016 ¦access-date=January 29, 2016}}&lt;/ref&gt; and [[Spike Lee]] declared they would boycott this year's Academy Awards ceremony, some of them attending [[Ryan Coogler]]'s charity event ''Justice for Flint'' instead.&lt;ref&gt;{{cite web ¦author=Lindsey Bahr ¦title=Black Celebrities Attend Justice For Flint Event Instead of (So White) Oscars ¦work=[[The Skanner]] ¦url=http://www.theskanner.com/entertainment/theater-movies/23582-black-celebrities-attend-justice-for-flint-event-instead-of-so-white-oscars ¦date=February 28, 2016 ¦access-date=February 28, 2016}}&lt;/ref&gt; Coogler however supported the nominated Sylvester Stallone to stand up for the film, attending the Academy Awards ceremony.&lt;ref name=LATimes&gt;{{cite news ¦author=Josh Rottenberg ¦title=Ryan Coogler and Ava DuVernay set to attend Flint benefit on Oscar night ¦newspaper=[[Los Angeles Times]] ¦url=http://www.latimes.com/entertainment/tv/showtracker/la-et-mn-ryan-coogler-ava-duvernay-flint-benefit-oscar-night-20160223-story.html ¦date=February 23, 2016 ¦access-date=February 29, 2016}}&lt;/ref&gt;

2894 ''Punch Club'' was initially developed by the Russia-based, three-person Lazy Bear Studios under the name ''VHS Story'', which stood for ''Video Hero Super Story''.&lt;ref name=&quot;siliconera interview&quot;&gt;{{Cite web ¦ url = http://www.siliconera.com/2015/02/06/interview-trading-blows-remembering-the-80s90s-with-the-vhs-story-creators/ ¦ title = Interview: Trading Blows, Remembering The 80s/90s With The VHS Story Creators ¦ first = Chris ¦ last = Priestman ¦ date = February 6, 2015 ¦ accessdate = February 23, 2016 ¦ work = [[Siliconera]] }}&lt;/ref&gt; Sviatoslav Cherkasov, one of the developers, said that their whole team, in 2015, were around thirty years old, and had grown up around [[Nintendo Entertainment System]] and [[Sega Genesis]] titles, cartoons like ''[[Teenage Mutant Ninja Turtles]]'', and action movies of the 80s and 90s distributed at that time on [[VHS tapes]] from stars like [[Arnold Schwarzenegger]], [[Sylvester Stallone]], and [[Jean-Claude Van Damme]].&lt;ref name=&quot;siliconera interview&quot;/&gt; As fans of strategy games, they sought to combine these features, coming up with what they considered to be a ''[[Street Fighter]]'' manager, inspired by the story of the ''Rocky'' movies.&lt;ref name=&quot;siliconera interview&quot;/&gt;

2895 ¦ subject = [[Sylvester Stallone]] as [[Rocky Balboa]]

2896 The '''Statue of Rocky Balboa''' ([[Serbo-Croatian language¦Serbo-Croatian]]: ''Статуа Рокија Балбое'', ''Statua Rokija Balboe'') is a bronze statue in [[Žitište]], [[Serbia]], dedicated to [[Rocky Balboa]], main character from 1976 American sports drama film ''[[Rocky]]'', portrayed by [[Sylvester Stallone]]. It was made by Croatian artist Boris Staparac, it was erected in 2007.&lt;ref&gt;{{cite web ¦url=http://www.b92.net/info/vesti/index.php?yyyy=2007&amp;mm=08&amp;dd=19&amp;nav_id=259816 ¦title= Otkriven spomenik Rokiju u Žitištu¦author=B92, Beta ¦date= 19 August 2007¦website= ¦ publisher= [[B92]]¦access-date=March 2, 2016 ¦quote=¦language= Serbian}}&lt;/ref&gt; Canadian director [[Barry Avrich]] made a [[documentary film]] ''Amerika Idol'' which depicts the events that preceded the creation of the statue, as well as the ceremony of its installation. The film also features [[Sylvester Stallone]] who portrayed the character of Rocky in eight films and [[A. Thomas Schomberg]] who made the famous [[Rocky Steps#Bronze Rocky statue¦''Rocky'' statue in Philadelphia]].&lt;ref&gt;{{cite web ¦url=http://www.blic.rs/zabava/vesti/film-o-rokiju-iz-zitista-u-trci-za-oskara/x5w2y9f ¦title= Film o Rokiju iz Žitišta u trci za &quot;Oskara&quot; ¦author=A. Novaković ¦date= 9 August 2009¦website= ¦publisher= [[Blic]]¦access-date=March 2, 2016 ¦quote=¦language= Serbian}}&lt;/ref&gt;

2897 {{Sylvester Stallone¦state=collapsed}}

2898 [[Category:Cultural depictions of Sylvester Stallone]]

2899 ¦ executive_producer = Dave Broome&lt;br&gt;Kevin King-Templeton&lt;br&gt;[[Sylvester Stallone]]&lt;br&gt;Holly Wofford

2900 '''''Strong''''' (stylized on-screen as '''''S.T.R.O.N.G.''''') was an American competition [[reality show]] that debuted on [[NBC]] on April 13, 2016 and aired on Thursdays at 8 p.m.&lt;ref name=&quot;Andreeva&quot;&gt;{{cite web ¦last=Andreeva ¦first=Nellie ¦title=Sylvester Stallone-Produced NBC Reality Series 'Strong' Gets Thursday Slot, Preview Behind 'The Voice' ¦ publisher=[[Deadline Hollywood]] ¦date=March 1, 2016 ¦url=https://deadline.com/2016/03/strong-premiere-date-sylvester-stallone-the-voice-nbc-1201712289/ ¦access-date=March 2, 2016}}&lt;/ref&gt; It was hosted by former professional volleyball player [[Gabrielle Reece]].&lt;ref&gt;{{cite news ¦last=Mazzo ¦first=Lauren ¦title=NBC's New Fitness Competition Show &quot;Strong&quot; Is All About the Sex Appeal ¦work=[[Shape (magazine)¦Shape]] ¦date=March 1, 2016 ¦url=http://www.shape.com/celebrities/star-trainers/nbcs-new-fitness-competition-show-strong-all-about-sex-appeal ¦access-date=March 2, 2016}}&lt;/ref&gt;

2901 Executive producers for ''Strong'' include Dave Broome, creator of the competition reality series ''[[The Biggest Loser (U.S. TV series)¦The Biggest Loser]],'' and Holly Wofford, a producer with the sports competition show ''[[American Ninja Warrior]]''. Other executive producers include [[Sylvester Stallone]] and Kevin King-Templeton, both of whom also collaborated on the sports drama film ''[[Creed (film)¦Creed]]'' (2015).&lt;ref name=&quot;Andreeva&quot; /&gt;&lt;ref name=&quot;BroadwayWorld&quot; /&gt; The series is being produced by [[Sony Pictures Television]] in association with 25/7 Productions.&lt;ref name=&quot;O'Connell&quot;&gt;{{cite news ¦last=O'Connell ¦first=Michael ¦title=NBC Nabs New Fitness Competition From 'Biggest Loser' Creator (Exclusive) ¦work=[[The Hollywood Reporter]] ¦date=March 12, 2015 ¦url=https://www.hollywoodreporter.com/live-feed/nbc-nabs-new-fitness-competition-780997 ¦access-date=March 2, 2016}}&lt;/ref&gt;

2902 Bode did the cinematography for films such as ''[[Saturday Night Fever]]'', ''[[Gorky Park (film)¦Gorky Park]]'', ''[[The Accused (1988 film)¦The Accused]]'', ''[[Uncle Buck]]'' and ''[[Don Juan DeMarco]]''. He was an uncredited second unit director for ''[[Rocky]]'' (1976), and is credited with shooting the scenes in which [[Sylvester Stallone]] runs up the steps of the Philadelphia Museum of Art. He received an [[Academy Award for Best Cinematography¦Academy Award]] nomination for his cinematography on ''Coal Miner's Daughter''. He was nominated for two [[Emmy Awards]] in the Outstanding Cinematography category: in [[46th Primetime Emmy Awards¦1994]] for ''[[Gypsy (1993 film)¦Gypsy]]'' and in [[52nd Primetime Emmy Awards¦2000]] for ''[[Annie (1999 film)¦Annie]]''.&lt;ref name=LAT/&gt;&lt;ref&gt;{{cite web ¦url=https://www.emmys.com/site-search?search_api_views_fulltext=bode&amp;f%5B0%5D=type%3Anominations ¦title=Bode nominations ¦author=&lt;!--Not stated--&gt; ¦ date=n.d. ¦website=emmys.com ¦publisher=Television Academy ¦access-date=June 12, 2022}}&lt;/ref&gt;

2903 &lt;li&gt;{{cite news ¦last=Levins ¦first=Harry ¦date=1993-08-01 ¦title=Clancy Looks Back in a Familiar Style ¦url=http://infoweb.newsbank.com/resources/doc/nb/news/0EB04E33F7871CF7?p=AWNB ¦ newspaper=[[St. Louis Post-Dispatch]] ¦accessdate=2016-04-04 ¦archiveurl=http://www.webcitation.org/6gVazRtrS ¦archivedate=2016-04-04 }}&lt;p&gt;The article notes:&lt;/p&gt; &lt;blockquote&gt;Clancy's regular readers may recall Clark as the grimly efficient CIA agent who has aided Ryan in three earlier novels. &quot;Without Remorse&quot; tells us who John Clark really is.&lt;p&gt;He's a cliche, actually. As the book opens, Clark lives as a recluse, a former Navy commando who saw too much combat and wants only to get away from the world. As is the case in any number of thrillers that star reclusive trained killers, the world refuses to go away.&lt;/p&gt;&lt;p&gt;In this book, the world presents Clark with two sets of problems: a heroin ring in Baltimore that murders his newfound lady friend, and a secret POW camp in North Vietnam. Clark responds by turning into two more cliches: Charles Bronson in those vigilante movies, and Sylvester Stallone in those Rambo movies.&lt;/p&gt;&lt;/blockquote&gt; The article provides analysis of Tom Clancy character John Clark by saying that he is a &quot;recluse&quot; who has seen &quot;too much combat and wants only to get away from the world&quot;. The article provides further analysis by comparing John Clark to [[Charles Bronson]] in &quot;vigilante movies&quot; and [[Sylvester Stallone]] in &quot;Rambo movies&quot;.&lt;/li&gt;

2904 ¦ {{sort¦ Stallone ¦ [[Sylvester Stallone]]}} ¦¦ M ¦¦ 1946 ¦¦ ~ ¦¦ {{Age¦ 1946 ¦7¦6}} ¦¦ 2 ¦¦ 0 ¦¦ 1L:1S ¦¦ ''[[Rocky]]'' ¦¦ ''1976'' ¦¦ 2015

2905 [[Sylvester Stallone]]

2906 [[File:John Rambo.jpg¦thumb¦right¦150px¦The film was heavily inspired by the [[Rambo (franchise)¦''Rambo'' franchise]] starring [[Sylvester Stallone]].]]

2907 In November 2015 it was reported that independent production company from Kuwait, Jabara Media Group, was developing an action movie which would share resemblances with American classics – ''[[First Blood]]'' (1982) and ''[[Rambo: First Blood Part II]]'' (1985). As indicated, Kuwaiti bodybuilding champion, police officer and real-life commando Abdulhadi Al-Khayat signed a contract to play Yousef Rambu, &quot;an ex-member of the Kuwait Army Forces, an exceptionally tough and courageous man.&quot;&lt;ref&gt;[http://www.fandimefilmu.cz/clanek/7163-rambo_chystaji_se_serial_a_kuvajtsky_remake &quot;Rambo: Chystají se seriál a kuvajtský remake&quot;]&lt;/ref&gt; Hailed as a Kuwaiti answer to [[John Rambo]], Rambu shares a lot of common characteristics with the American movie icon.&lt;ref&gt;[http://kulturistika.ronnie.cz/c-23651-kuvajtsky-profesional-abdulhadi-al-khayat-fotografovani-v-posilovne.html &quot;Kuvajtský profesionál Abdulhadi Al-Khayat: fotografování v posilovně&quot;]&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=https://primemovies.pl/2022/07/28/aktorzy-ktorzy-mieli-zagrac-rambo/¦title=Ten actors that almost portrayed John Rambo on screen¦work=Prime Movies¦date=July 28, 2022¦access-date=July 29, 2022}}&lt;/ref&gt; Al-Khayat's appearance in ''Second Blood'' is his first movie performance, and the movie itself is a screenwriting debut for both Shehab Al-Fadhli and Fayez Hussein Ali. In late December 2015 it was announced that Fawzi Al-Khatib will direct the project.&lt;ref name=&quot;mr25&quot;&gt;{{Cite web ¦url=http://moviesroom.pl/kuwejcki-film-inspirowany-seria-rambo-zyskuje-rezysera/ ¦title=&quot;Kuwejcki film inspirowany serią &quot;Rambo&quot; zyskuje reżysera&quot; ¦access-date=2016-05-16 ¦ archive-date=2016-01-05 ¦archive-url=https://web.archive.org/web/20160105030359/http://moviesroom.pl/kuwejcki-film-inspirowany-seria-rambo-zyskuje-rezysera/ ¦url-status=dead }}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦url=http://www.ppe.pl/rozrywka/2094/kacik-filmowy-nowy-zwiastun-deadpoola-przebudzenie-mocy-alien-covenant-i-duzo-wiecej.html ¦title=&quot;Kącik filmowy: nowy zwiastun Deadpoola, Przebudzenie mocy, Alien: Covenant i dużo więcej&quot; ¦access-date=2016-05-16 ¦archive-date=2016-03-05 ¦archive-url=https://web.archive.org/web/20160305033121/http://www.ppe.pl/rozrywka/2094/kacik-filmowy-nowy-zwiastun-deadpoola-przebudzenie-mocy-alien-covenant-i-duzo-wiecej.html ¦url-status=dead }}&lt;/ref&gt; Al-Khatib found Rambu to be &quot;an improved version of the original Rambo.&quot;&lt;ref name=&quot;mr30&quot;/&gt; It was also reported that Ranaa Ghandour will play the role of Rambu's wife, Monira.&lt;ref name=&quot;mr25&quot;/&gt; Filming commenced in February 2016 in [[Punjab (region)¦Punjab]], [[Pakistan]].&lt;ref name=&quot;fb&quot;/&gt;&lt;ref&gt;[http://www.kinobox.cz/clanek/11029-sylvester-stallone-vyviji-pro-fox-serial-rambo-nova-krev &quot;Sylvester Stallone vyvíjí pro Fox seriál Rambo: Nová krev. A chystá se i remake původního filmu!&quot;]&lt;/ref&gt; Completion of the material continued until May.&lt;ref name=&quot;mr30&quot;/&gt;

2908 <title>Awards and nominations received by Sylvester Stallone</title>

2909 <redirect title="List of awards and nominations received by Sylvester Stallone" />

2910 <comment>/* top */Redirecting to [[List of awards and nominations received by Sylvester Stallone]] using [[Project:AWB¦AWB]]</comment>

2911 <text bytes="97" sha1="ncf6q3tiqgpocae3lbztzh8cp4xz8nl" xml:space="preserve">#REDIRECT [[List of awards and nominations received by Sylvester Stallone]]

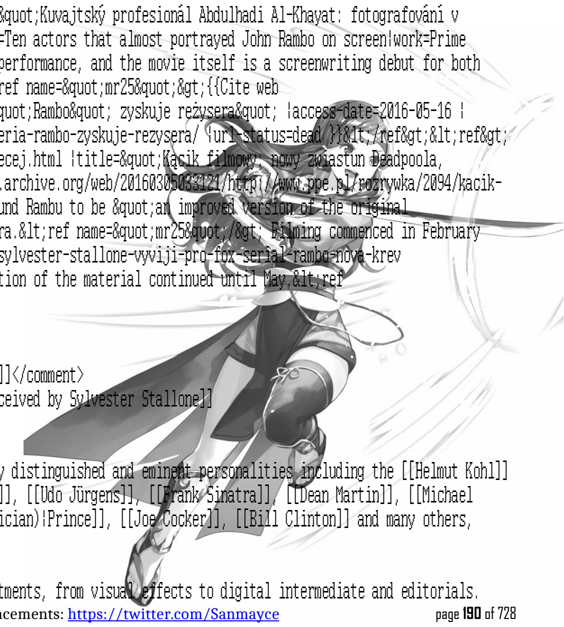
2912 * [[Sylvester Stallone]] - ''[[Creed (film)¦Creed]]''

2913 ¦ [[Sylvester Stallone]]

2914 Alex's presence and personality attracted representatives of the business, politics, nobility, culture and science elite and many distinguished and eminent personalities including the [[Helmut Kohl]] family, Prince of Monaco, Gilles Hennessy, [[Ephraim Kishon]], [[Sylvester Stallone]], [[Claude Brasseur]], [[Jean Paul Belmondo]], [[Udo Jürgens]], [[Frank Sinatra]], [[Dean Martin]], [[Michael Jackson]], [[Lino Ventura]], [[The Rolling Stones]], [[Ella Fitzgerald]] and [[Stevie Wonder]], [[Benny Goodman]], [[Prince (musician)¦Prince]], [[Joe Cocker]], [[Bill Clinton]] and many others, happily sang alongside to Alex's piano accompaniment.&lt;br /&gt;

2915 # [[Sylvester Stallone]] &amp;ndash; ''[[Creed (film)¦Creed]]'' (14)

2916 From 2004 to 2009, Bagdasarian worked on over a hundred feature films, in a variety of film production and post-production departments, from visual effects to digital intermediate and editorials.



```
Bagdasarian was credited on such films as [[Sylvester Stallone]]'s &quot;[[Rambo (2008 film)|Rambo]]&quot; and [[Werner Herzog]]'s &quot;[[Bad Lieutenant: Port of Call New Orleans|Bad Lieutenant]].&quot;
2917 Scenes in several other movies, including ''[[Rocky II]]'' (1979) starring [[Sylvester Stallone]], ''[[Mannequin (1987 film)|Mannequin]]'' (1987) starring [[Andrew McCarthy]] and [[Kim Cattrall]], ''[[In Her Shoes (film)|In Her Shoes]]'' (2005) starring [[Cameron Diaz]], [[Shirley MacLaine]], and [[Toni Collette]], and ''[[The Happening (2008 film)|The Happening]]'' (2008) starring [[Mark Wahlberg]], were filmed a few streets away in nearby Rittenhouse Square. The exceptional use of this one city block by the movie and television industry relates to the history and aesthetics of Delancey Place, and cultural setting as an idyllic city street. For example, in ''[[Trading Places]]'', it was used to convey the great wealth of the main character while in ''[[The Answer Man]],'' the aesthetic of the street was used in a strolling scene,&lt;ref&gt;Paetzold, Heinz (2013). The aesthetics of city strolling. Contemporary Aesthetics vol. 11, p.10.&lt;/ref&gt; where the two main characters were interacting.
2918 [[Rocky Balboa]], from the [[Rocky (film series)|''Rocky'' film series]], was chosen as the inaugural induction to start the Fictitious Athlete Hall of Fame.&lt;ref&gt;{{cite web|last1=Greene|first1=Jerry|title=Fictitious Hall has a lot of character(s)|url=http://www.orlandosentinel.com/sports/os-jerry-greene-cheap-seats-0503-20150502-column.html|website=Orlando Sentinel|publisher=[[Orlando Sentinel]]|date=May 2, 2015}}&lt;/ref&gt; Buchner stated that the very inspiration for creating the page was seeing outrage at [[Sylvester Stallone]] being inducted into the [[International Boxing Hall of Fame]] for Rocky.&lt;ref name=radley/&gt;
2919 Sheridan got his start portraying Danny Boyd in ''[[Veronica Mars]]'' (2005-2007) as well as portraying [[David Hale (Sons of Anarchy)|David Hale]] in the [[FX (TV channel)|FX]] television series ''[[Sons of Anarchy]]''.  He also co-created the crime thriller ''[[Mayor of Kingstown]]'', and created the crime drama ''[[Tulsa King]]'', which he co-writes with [[Terence Winter]].&lt;ref&gt;{{cite web |last1=White |first1=Peter |title=Sylvester Stallone Set To Star In Taylor Sheridan &amp; Terence Winter Drama Series 'Kansas City' For Paramount+ From 101 Studios &amp; MTV Ent. Studios |url=https://deadline.com/2021/12/sylvester-stallone-taylor-sheridan-terence-winter-drama-series-kansas-city-paramount-1234883070/ |website=Deadline Hollywood |publisher=Penske Media Corporation |access-date=December 12, 2021 |language=English |date=December 6, 2021 }}&lt;/ref&gt;
2920 The video begins at 7th and [[Broadway (Los Angeles)|Broadway]] in [[Downtown Los Angeles]], one block over from the site of [[Avril Lavigne]]'s infamous street concert in 2002 video [[Sk8er Boi]]. Then it cuts to the band playing on a rooftop intercut with shots of the band escaping from popular [[internet memes]], as well as internet public figures, that are attacking them, and then ultimately fighting them. Among the memes and public figures in the videos are the [[Nyan Cat]], [[Kim Kardashian]] posing nude (from her 'Break the Internet' photo shoot for [[Paper (magazine)|Paper]] magazine), [[Miley Cyrus]] on a [[Wrecking Ball (Miley Cyrus song)|Wrecking Ball]], [[Sylvester Stallone]] as ''[[John Rambo]]'', [[Angry Birds]], [[Doge (meme)|Doge]], [[Bill Cosby]], [[Brad Pitt]], [[Charlie Sheen]], [[Sean Bean]], [[Ben Affleck]],  [[Donald Trump]], [[Bernie Sanders]], [[James Franco]], [[Seth Rogen]], [[Michael Jordan]], [[Shia LaBeouf]] from [[LaBeouf, Rönkkö &amp; Turner#.23INTRODUCTIONS.2C 2015|his motivational video]], a few [[Guitar Hero]] references (like [[Deryck Whibley]] playing a Guitar Hero game controller instead of his actual guitar for a few seconds), [[Trevor Philips]] from ''[[Grand Theft Auto V]]'', [[Kermit the Frog]], [[SpongeBob SquarePants (character)|SpongeBob]], [[Spider-Man]] and more. It is the first video by the band to feature the return of guitarist [[Dave Baksh]] and the first video not to feature former drummer [[Steve Jocz]], who was replaced by [[Frank Zummo]].
2921 ** Kroll also voices Mila and Lotte Janssen, the [[Statue of Liberty]], a profane ladybug, [[Joe Walsh]], the ghosts of [[Pablo Picasso|Picasso]] and [[Richard Burton]], [[Sylvester Stallone]], a webcam girl, several tampons, Gina's Abuela, Bad Mitten and Rabbi Poblart.
2922 Throughout most of the 1980s and 90s, Rothe had about ten large shows each year in different US cities, and Peter regularly showed her work at the [[Art Basel|Basel Art Fair]]. Every month or two, she created a new edition for galleries to buy. She made all of her work with the help of one assistant. Rothe also did paintings for private commissions, but she rarely kept track of these sales. [[Sylvester Stallone]] and [[Clint Eastwood]], who were her neighbors in Carmel, both own some of her paintings. She painted a large outdoor mural in the Hog's Breath Inn, Clint Eastwood's former restaurant in Carmel.
2923 *... that, inspired by [[Sylvester Stallone]]'s experience selling the script for ''[[Rocky]]'', actor/screenwriter [[J. P. Davis]] refused to sell his script for the film '''''[[Fighting Tommy Riley]]''''' unless he was guaranteed to play the lead? &lt;small&gt;(2008-03-31)&lt;/small&gt;
2924 In 2016, [[STX Entertainment]] and Grey Matter Productions announced an upcoming thriller starring [[Sylvester Stallone]]. The film, written by Dan Casey and to be directed by [[Jim Mickle]], will go into production in late 2016.&lt;ref&gt;{{cite web|last1=Fleming|first1=Mike Jr.|title=Jim Mickle To Direct Sylvester Stallone In STX Entertainment Thriller|url=https://deadline.com/2016/06/sylvester-stallone-jim-mickle-director-stx-entertainment-drama-1201772054/|website=deadline.com|access-date=13 June 2016}}&lt;/ref&gt; Grey Matter also announced that it will produce a feature film about the recent [[Panama Papers]] scandal.  [[Steven Soderbergh]] is also producing with an eye to direct, alongside [[Scott Z. Burns]]. The film is to be based on an upcoming book ''Secrecy World'' that [[Pulitzer Prize]]-winning journalist [[Jake Bernstein (journalist)|Jake Bernstein]] is writing for [[Henry Holt and Company]]. Grey is teamed with [[Anonymous Content]] to develop, finance and produce. Grey, [[Michael Sugar]], Burns and Soderbergh are producing.&lt;ref&gt;{{cite web|last1=Fleming|first1=Mike Jr.|title=Steven Soderbergh Plots Panama Papers Film Based On Jake Bernstein Book 'Secrecy World'|url=https://deadline.com/2016/07/steven-soderbergh-panama-papers-movie-jake-bernstein-secrecy-world-scott-burns-1201783424/|website=deadline.com|access-date=6 July 2016}}&lt;/ref&gt;
2925 | ''[[The Expendables (2010 film)|The Expendables]]'' || [[Sylvester Stallone]] || Brian Tyler
2926 *{{icon|DYK}} ... that, inspired by [[Sylvester Stallone]]'s experience selling the script for ''[[Rocky]]'', actor/screenwriter [[J. P. Davis]] refused to sell his script for the film '''''[[Fighting Tommy Riley]]''''' unless he was guaranteed to play the lead? &lt;small&gt;(2008-03-31)&lt;/small&gt;
2927 * [[Sylvester Stallone]]
2928 '''''Guardians of the Galaxy Vol. 3'''''&lt;!-- This is the title according to the billing block from the poster as well as official press releases --&gt; (marketed as '''''Guardians of the Galaxy Volume 3''''') is a 2023 American [[superhero film]]&lt;!-- WP:FILMLEAD, only primary genre or sub-genre --&gt; based on the [[Marvel Comics]] superhero team [[Guardians of the Galaxy (2008 team)|Guardians of the Galaxy]], produced by [[Marvel Studios]] and distributed by [[Walt Disney Studios Motion Pictures]]. It is the sequel to ''[[Guardians of the Galaxy (film)|Guardians of the Galaxy]]'' (2014) and ''[[Guardians of the Galaxy Vol. 2]]'' (2017), and [[List of Marvel Cinematic Universe films|the 32nd film]] in the [[Marvel Cinematic Universe]] (MCU). Written and directed by [[James Gunn]], it features an [[ensemble cast]] including [[Chris Pratt]], [[Zoe Saldaña]], [[Dave Bautista]], [[Karen Gillan]], [[Pom Klementieff]], [[Vin Diesel]], [[Bradley Cooper]], [[Will Poulter]], [[Sean Gunn]], [[Chukwudi Iwuji]], [[Linda Cardellini]], [[Nathan Fillion]], and [[Sylvester Stallone]].&lt;!--Cast billing based the film's main titles; see [[WP:MCUFILMCAST]].--&gt; In the film, the other [[Guardians of the Galaxy (Marvel Cinematic Universe)|Guardians]] race to save the life of [[Rocket (Marvel Cinematic Universe)|Rocket]] (Cooper) from his creator, the [[High Evolutionary (Marvel Cinematic Universe)|High Evolutionary]] (Iwuji), who is an alien scientist trying to perfect the universe.
2929 * [[Sylvester Stallone]] as [[Stakar Ogord (Marvel Cinematic Universe)|Stakar Ogord]]: A high-ranking Ravager.&lt;ref name=&quot;Stallone&quot; /&gt;
2930 [[Principal photography]] began on November 8, 2021,&lt;ref name=&quot;FilmingStart&quot; /&gt; at [[Trilith Studios]] in Atlanta, Georgia,&lt;ref name=&quot;CastingFilming2&quot; /&gt;&lt;ref name=&quot;TrilithFilming&quot; /&gt; under the [[working title]] ''Hot Christmas''.&lt;ref name=&quot;2021Production&quot; /&gt; [[Henry Braham]] served as cinematographer, after doing so for ''Vol. 2'', ''The Suicide Squad'',  and ''The Guardians of the Galaxy Holiday Special''.&lt;ref name=&quot;Braham&quot; /&gt; Filming was previously scheduled to begin in January or February 2019 prior to Gunn's firing,&lt;ref name=&quot;Postponement&quot; /&gt;&lt;ref name=&quot;Jan2019Filming&quot; /&gt; and then in February 2021,&lt;ref name=&quot;2021Production&quot; /&gt; before Gunn began work on ''Peacemaker''.&lt;ref name=&quot;Gunn2021Production&quot; /&gt; With the start of filming, [[Sylvester Stallone]] revealed that he would return as [[Stakar Ogord (Marvel Cinematic Universe)|Stakar Ogord]] from ''Vol. 2'',&lt;ref name=&quot;Stallone&quot; /&gt; and Gunn posted a photo of the main cast members which revealed that [[Chukwudi Iwuji]] was part of the film following his collaboration with Gunn on ''Peacemaker''.&lt;ref name=&quot;FilmingStart&quot; /&gt;&lt;ref name=&quot;IwujiCast&quot; /&gt; Iwuji's screen test for the film was shot on the set of ''Peacemaker'' with that series' crew, and Marvel repaid this favor by letting Gunn use the ''Vol. 3'' set and crew to film [[Ezra Miller]]'s cameo appearance as [[Barry Allen (DC Extended Universe)|Barry Allen / The Flash]] for the ''Peacemaker'' [[It's Cow or Never|season finale]].&lt;ref name=&quot;GunnVarietyFeb2022&quot; /&gt;
2931 &lt;ref name=&quot;Stallone&quot;&gt;{{Cite web |last=DeVore |first=Britta |date=November 12, 2021 |title=Sylvester Stallone Confirms His Return to the MCU in 'Guardians of the Galaxy Vol. 3' |url=https://collider.com/guardians-of-the-galaxy-3-sylvester-stallone/ |url-status=live |archive-url=https://web.archive.org/web/20211113035650/https://collider.com/guardians-of-the-galaxy-3-sylvester-stallone/ |archive-date=November 13, 2021 |access-date=November 12, 2021 |website=[[Collider (website)|Collider]]}}&lt;/ref&gt;
2932 &lt;ref name=&quot;Fillion&quot;&gt;{{Cite web |last=McCall |first=Kevin |date=April 12, 2023 |title=Sylvester Stallone and Nathan Fillion Hijack the New 'Guardians of the Galaxy Vol. 3' Trailer |url=https://collider.com/guardians-of-the-galaxy-vol-3-trailer-sylvester-stallone-nathan-fillion/ |url-status=live |archive-url=https://web.archive.org/web/20230423104145/https://collider.com/guardians-of-the-galaxy-vol-3-trailer-sylvester-stallone-nathan-fillion/ |archive-date=April 23, 2023 |access-date=April 13, 2023 |website=[[Collider (website)|Collider]]}}&lt;/ref&gt;
2933 ''Six Million Ways to Live'' a song co-written by O'Bryan was used as the end title music in the [[Chris Rock]] and [[Anthony Hopkins]] movie Bad Company and Dub Pistols song ''Official Chemical'' was used in the Joel Schumacher movie 'Driven' featuring [[Sylvester Stallone]]. The US series ''Bullrun'' also features O'Bryan's music as the programme's theme music as well as incidental music throughout the series.
2934 Stamp appeared with several other professional wrestlers, including [[Ted DiBiase]], [[Bob Roop]], [[Dick Murdoch]], [[Gene Kiniski]], and [[Tonga Fifita]], in the 1978 [[Sylvester Stallone]] movie ''[[Paradise Alley]]''.&lt;ref name=&quot;pwi&quot;/&gt;
2935 The remake was first announced in 2006, with [[Sylvester Stallone]] set to direct and star in the lead role, before dropping out. Carnahan was announced to write and direct in 2012, but later dropped out although was given sole screenwriting credit despite his script being rewritten numerous times. Roth joined as director in 2016, with filming taking place in [[Chicago]] that same year with director of photography [[Rogier Stoffers]]. [[Mark Goldblatt]] served as editor in post-production, while the score was composed by [[Ludwig Göransson]].
2936 Development on the film began already back in 2006, when [[Sylvester Stallone]] announced that he would be directing and starring in a remake of ''[[Death Wish (1974 film)|Death Wish]]'' (1974). Stallone told [[Ain't It Cool News]], &quot;Instead of the [[Charles Bronson]] character being an architect, my version would have him as a very good cop who had incredible success without ever using his gun. So when the attack on his family happens, he's really thrown into a moral dilemma in proceeding to carry out his revenge.&quot; He later told the publication that he was no longer involved after &quot;creative differences&quot; and became busy with work on other projects.&lt;ref name=&quot;MHoleRemake&quot;&gt;{{cite news |first=Clint |last=Morris |author-link=Clint Morris |title=Stallone in Death Wish remake? Updated! |url=http://www.moviehole.net/news/20061220_stallone_in_death_wish_remake.html |work=Moviehole |date=June 8, 2008 |access-date=December 20, 2006 |quote=Upon listening to the talkback responses on AICN, many who turned their nose at the idea of a remake, Stallone tells the site today that he will NOT be doing the movie. Yep, he listened to the fans! |url-status=dead |archive-url=https://web.archive.org/web/20070703060619/http://www.moviehole.net/news/20061220_stallone_in_death_wish_remake.html |archive-date=July 3, 2007 }}&lt;/ref&gt;&lt;ref&gt;{{cite news|title=Stallone tapped for Death Wish remake|url=http://film.guardian.co.uk/news/story/0,,2205587,00.html|work=[[The Guardian]]|publisher=[[Guardian News and Media]]|access-date=November 5, 2007 | date=November 5, 2007 |author=Staff and agencies}}&lt;/ref&gt;
2937 In a 2009 interview with [[MTV]], Stallone stated that he was again considering the project.&lt;ref&gt;{{cite web|last1=Rosenberg|first1=Adam|title=Sylvester Stallone Speaks on a 'Death Wish' Remake and Edgar Allen Poe|url=http://www.mtv.com/news/2433850/sylvester-stallone-speaks-on-a-death-wish-remake-and-edgar-allen-poe/|archive-url=https://web.archive.org/web/20180221100840/http://www.mtv.com/news/2433850/sylvester-stallone-speaks-on-a-death-wish-remake-and-edgar-allen-poe/|url-status=dead|archive-date=February 21, 2018|date=October 2, 2009|publisher=[[MTV]]|work=[[MTV News]]|
```

access-date=February 21, 2018}}&lt;/ref&gt; In late January 2012, ''[[The Hollywood Reporter]]'' confirmed that a remake would be written and directed by [[Joe Carnahan]].&lt;ref&gt;{{cite web| title = 'The Grey' Director Joe Carnahan to Remake 'Death Wish'| url = https://www.hollywoodreporter.com/news/the-grey-liam-neeson-joe-carnahan-charles-bronson-death-wish-remake-286136|work=[[The Hollywood Reporter]]|publisher=[[Prometheus Global Media]] |access-date = October 5, 2012 |last=Gilchrist|first=Todd|date=January 31, 2012}}&lt;/ref&gt; The film was originally set to star [[Liam Neeson]] and [[Frank Grillo]]. Carnahan too left the project in February 2013 due to &quot;creative differences&quot;,&lt;ref name=&quot;desde&quot;&gt;{{cite web|title=Exclusive Benicio Del Toro Rejected to Star in Death Wish Remake 1|url=http://www.desdehollywood.com/exclusive-benicio-del-toro-rejected-lead-death-remake/|work=Desde Hollywood|access-date=February 21, 2018|last=Bentancor|first=Nestor|date=May 21, 2013 }}&lt;/ref&gt; but received sole writing credit for the completed film.&lt;ref&gt;{{cite web|last1=Sprague|first1=Mike|title=Bruce is Back in Bitchin' New Retro Poster for Eli Roth's Death Wish Remake|url=https://www.dreadcentral.com/news/267271/bruce-back-bitchin-new-retro-poster-eli-roths-death-wish-remake/|work=[[Dread Central]]|publisher=Dread Central Media|access-date=February 26, 2018|date=February 20, 2018}}&lt;/ref&gt; He was replaced as director with Gerardo Naranjo, who was interested in casting [[Benicio Del Toro]] in the lead role; this version also never came to fruition.&lt;ref name=&quot;desde&quot;/&gt;

2938 | ShortSummary   = Contestants: &lt;br&gt;'''Cynthia - [[Sylvester Stallone]] expert'''&lt;br&gt;Heather - [[Labyrinth]]s expert&lt;br&gt;Aaron - [[Leeds United F.C.]] expert&lt;br&gt;Daley - [[Rat]]s expert

2939 '''[[Sylvester Stallone]]'''

2940 | executive_producer   = {{ubl|Dave Broome|[[Sylvester Stallone]]|Yong Yam|Kevin King-Templeton|Elayne Cilic|Travis McDaniel|Mike Espinosa}}

2941 '''''Ultimate Beastmaster''''' is an American [[Reality television|reality]] [[sports entertainment]] competition that premiered on [[Netflix]] on February 24, 2017. The show differentiates itself from other [[obstacle course]] competition shows by showcasing not only international talent, but by producing six localized versions, featuring select television hosts, actors, comedians and athletes as commentators from the six countries competing in the show. Reactions from all countries' hosts appear in each localized versions, especially during course clearing victories. Each season consists of 10 episodes, released simultaneously on Netflix worldwide.&lt;ref&gt;{{cite web|url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|title=Netflix Orders Global Competition Series 'Ultimate Beastmaster' from Sylvester Stallone, Dave Broome|last=Prudon|first=Laura|website=Variety|date=May 9, 2016|access-date=October 5, 2016|archive-date=November 8, 2020|archive-url=https://web.archive.org/web/20201108100911/https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|url-status=live}}&lt;/ref&gt; The first season premiered on February 24, 2017, while the second season aired on Netflix on December 15, 2017.

2942 In each episode, there are twelve contestants (two from each country) who run a new obstacle course known as "The Beast". The winner is crowned "Beastmaster". Each of the &quot;Beastmasters&quot; from the nine Beastmaster episodes will go forward to a final course, to become the Ultimate Beastmaster.&lt;ref name=details&gt;{{cite web|url=https://deadline.com/2016/05/netflix-ultimate-beastmaster-sylvester-stallone-competition-reality-series-1201751580/|title=Netflix Orders 'Ultimate Beastmaster' Competition Reality Series With Sylvester Stallone|last=Andreeva|first=Nellie|website=Deadline|date=May 9, 2016|access-date=October 5, 2016|archive-date=September 29, 2017|archive-url=https://web.archive.org/web/20170929000334/https://deadline.com/2016/05/netflix-ultimate-beastmaster-sylvester-stallone-competition-reality-series-1201751580/|url-status=live}}&lt;/ref&gt; Starting in Season 3, each show featured nine competitors (one from each country in the competition). The two competitors with the highest score in the final stage moved onto one of two semi-finals. The top three from each semi-final moved to the final episode to be crowned Ultimate Beastmaster.

2943 In his down-time Flowers enjoys participating in the award-winning Brio [[show choir]] and his church youth group. He also enjoys hiking, singing, dancing, playing piano, writing songs, creating his own stop-motion [[Lego]] movies, and watching professional wrestling. His favorite color is blue and the actor he'd most like to work with is [[Sylvester Stallone]].&lt;ref&gt;{{Cite web|url=http://starsqa.com/aiden-flowers-interviews|title=Aiden Flowers - Interviews {{!}} StarsQA|last=Hills|first=Jenny|website=starsqa.com|access-date=2016-10-05}}&lt;/ref&gt;

2944 Before the fight, Wepner was lying in bed with his wife when he told her something like &quot;Even if I don't win, I just want to prove I belong there.&quot; Wepner later shared this line with [[Sylvester Stallone]] who used it in ''[[Rocky]]''.&lt;ref name = &quot;Facing Ali 221&quot;&gt;{{cite book|title=Facing Ali|author=Stephen Brunt |publisher=The Lyons Press|page = 221|year=2002}}&lt;/ref&gt; On the day of the fight Wepner presented a &quot;very sexy&quot; blue negligee to his wife and told her to wear it that night in bed since she would be sleeping with the heavyweight champion of the world. On the night of the fight, Mrs. Wepner was wearing the negligee when Wepner returned to their hotel room, after the fight, with twenty three stitches. She asked him: &quot;Okay, bigshot...Do I go to Ali's room, or does he come to mine?&quot;&lt;ref name = &quot;Facing Ali 221-2&quot;&gt;{{cite book|title=Facing Ali|author=Stephen Brunt |publisher=The Lyons Press|pages = 221-2|year=2002}}&lt;/ref&gt;&lt;ref name = &quot;Hauser 302&quot;&gt;{{cite book|title=Muhammad Ali:His Life and Times|author= Thomas Hauser|publisher=Simon &amp; Schuster|page = 302|year=1991}}&lt;/ref&gt;

2945 The fight was sufficiently inspiring for Sylvester Stallone (who watched the fight live) to partly base the character of [[Rocky Balboa]] on Wepner, and of [[Apollo Creed]] on Ali. Stallone would later rush home after the fight and in less than four days of near constant writing, had completed his screenplay about an underdog fighter who gets a shot at the heavyweight title. The screenplay, titled ''Rocky,'' would be purchased by [[United Artists]], with the agreement that Stallone would also star in the film. Released the following year, ''Rocky'' would go on to win three [[Academy Awards|Oscars]] (including [[Academy Award for Best Picture|Best Picture]]) and became the highest-grossing film of [[1976 in film|1976]].

2946 | author = |30px|30px|Sylvester Stallone&lt;ref&gt;{{cite web|url=http://www.totalrocky.com/films/rocky/production.shtml |title=Rocky: Behind the Scenes Rocky (1976) &amp;#124; Making Of Rocky |website=Totalrocky.com |date=2014-06-20 |access-date=2015-12-01}}&lt;/ref&gt;

2947 }}At the [[49th Academy Awards|1977 Academy Awards]], Stallone and Ali presented the [[Academy Award for Best Supporting Actress]] together, with Ali comedically announcing that he was the real Apollo Creed and mock sparring with Stallone.&lt;ref&gt;{{Citation |title=Sylvester Stallone and Muhammad Ali Face-Off at the Oscars |url=https://www.youtube.com/watch?v=R-nd39yl5BU |access-date=2023-12-18 |language=en}}&lt;/ref&gt;

2948 Among the celebrities included in the film are the [[Arctic Monkeys]], [[Benedict Cumberbatch]], [[Bridgit Mendler]], [[Carly Rae Jepsen]], [[Justin Bieber]], [[Metallica]], [[Miss Piggy]], [[Morgan Freeman]], [[Sylvester Stallone]], [[Steve-O]], and [[Tommy Chong]].&lt;ref name=stereogum /&gt;&lt;ref&gt;{{cite news |last=Falkner |first=Scott |date=22 December 2014 |title=Lennon or McCartney? New Documentary Asks 550 Celebrities Their Preference-See Their Answers |url=http://www.inquisitr.com/1697375/lennon-or-mccartney-new-documentary-asks-550-celebrities-their-preference-see-their-answers/ |publisher=[[Inquisitr]] |access-date=11 November 2016 }}&lt;/ref&gt; Of the responses, 282 answered &quot;[[John Lennon]]&quot;, 196 answered &quot;[[Paul McCartney]]&quot;, 15 answered &quot;[[George Harrison]]&quot;, four answered &quot;[[Ringo Starr]]&quot;, one answered &quot;[[Jimi Hendrix]]&quot;, one answered &quot;[[Lou Reed]]&quot;, one answered &quot;[[Keith Richards]]&quot;, one answered &quot;[[Oasis (band)|Oasis]]&quot;, and 50 did not answer.&lt;ref name=houston&gt;{{cite news |last=Hoffman |first=Ken |date=26 December 2014 |title=Lennon or McCartney? In one-word answers |url=http://www.houstonchronicle.com/life/columnists/hoffman/article/Lennon-or-McCartney-In-one-word-answers-5980140.php |newspaper=[[Houston Chronicle]] |access-date=11 November 2016 }}&lt;/ref&gt;

2949 * [[Robert De Niro]] and [[Sylvester Stallone]]

2950 In 1984, Cameron wrote a first draft under the title ''[[Rambo: First Blood Part II|First Blood II]]''. (Cameron had been recommended by [[David Giler]] who did some uncredited script work on the first film.&lt;ref&gt;{{cite news|author=Broeske, P. H.|date=Oct 27, 1985|title=THE CURIOUS EVOLUTION OF JOHN RAMBO|work=Los Angeles Times|id={{ProQuest|154252710}}}}&lt;/ref&gt;) Cameron's script had the same basic structure of the first film but had the character of Rambo's partner before later drafts were written by star [[Sylvester Stallone]]. Stallone's final draft differed radically from Cameron's initial vision.&lt;ref&gt;{{cite web|url=http://www.amazingcameron.com/biography.html |title=Biography |publisher=Amazingcameron.com |access-date=August 27, 2010 |url-status=dead |archive-url=https://web.archive.org/web/20100725030036/http://www.amazingcameron.com/biography.html |archive-date=July 25, 2010 }}&lt;/ref&gt;

2951 :* [[Sylvester Stallone]] - ''[[Creed (film)|Creed]]''

2952 |[[Sylvester Stallone]]

2953 MediaFetcher.com is a fake news website generator. It has various templates for creating false articles about celebrities of a user's choice. Often users miss the disclaimer at the bottom of the page, before re-sharing. The website has prompted many readers to speculate about the deaths of various celebrities.&lt;ref&gt;{{Cite news|url=https://www.snopes.com/fact-check/celebrity-car-crash-hoax/|title=Joe Montana Killed in Car Accident|last=Mikkelson|first=David|work=Snopes.com|access-date=2020-09-13|language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=https://www.snopes.com/fact-check/sylvester-stallone-death-hoax/|title=Sylvester Stallone Death Hoax|work=Snopes.com|access-date=2020-09-13|language=en-US|last=David|first=Mikkelson}}&lt;/ref&gt;

2954 |[[Sylvester Stallone]] ‡ || ''[[Creed (film)|Creed]]'' || [[Rocky Balboa]]

2955 | [[Michael France]], [[Sylvester Stallone]]

2956 * The film poster is a parody of ''[[First Blood]]'', where Gérard Jugnot appears at the middle of the picture with torn clothes, a band on the cap and holding an [[M60 machine gun]] like [[Sylvester Stallone]].

2957 | [[Sylvester Stallone]]

2958 | [[Sylvester Stallone]]

2959 | [[Sylvester Stallone]]

2960 * '''[[Sylvester Stallone]] - ''[[Creed (film)|Creed]]'''''

2961 * [[Sylvester Stallone]] (1986)

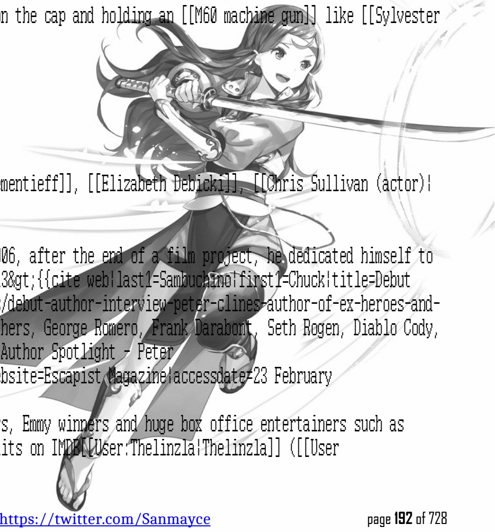
2962 | [[Chris Pratt]], [[Zoe Saldana]], [[Dave Bautista]], [[Vin Diesel]], [[Bradley Cooper]], [[Michael Rooker]], [[Karen Gillan]], [[Pom Klementieff]], [[Elizabeth Debicki]], [[Chris Sullivan (actor)|Chris Sullivan]], [[Sean Gunn]], [[Sylvester Stallone]] and [[Kurt Russell]]

2963 | [[Sylvester Stallone]] || {{USA}}

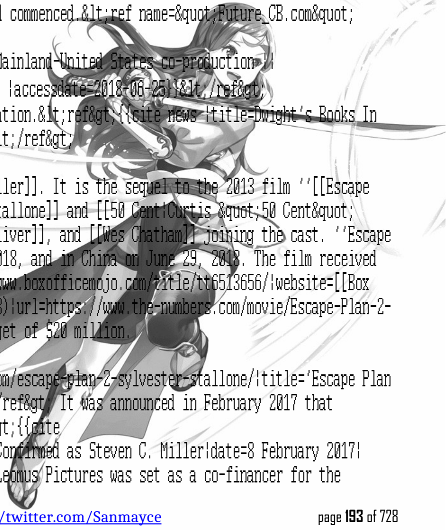
2964 While Clines worked as a props master, his writing turned from props to focus on scripts.&lt;ref name=&quot;sagliani2014&quot;/&gt; In 2006, after the end of a film project, he dedicated himself to writing full-time. He worked for ''Creative Screenwriting Magazine'', writing interviews, reviews, and articles.&lt;ref name=Sambuchino2013&gt;{{cite web|last1=Sambuchino|first1=Chuck|title=Debut Author Interview: Peter Clines, Author of EX-HEROES and EX-PATRIOTS|url=http://www.writersdigest.com/editor-blogs/guide-to-literary-agents/debut-author-interview-peter-clines-author-of-ex-heroes-and-ex-patriots|website=Writer's Digest|accessdate=21 February 2017}}&lt;/ref&gt; While at Creative Screenwriter, Clines interviewed, among others, George Romero, Frank Darabont, Seth Rogen, Diablo Cody, Sylvester Stallone, and the late Nora Ephron.&lt;ref name=&quot;sagliani2014&quot;&gt;{{cite web|last1=Sagliani|first1=Devan|title=Horror Author Spotlight - Peter Clines|url=http://www.escapistmagazine.com/articles/view/comicsandcosplay/columns/darkdreams/12004-Horror-Author-Spotlight-Peter-Clines|website=Escapist Magazine|accessdate=23 February 2017}}&lt;/ref&gt;

2965 Mr. Lindsay has been involved in over 20 movies and TV shows. Including in these he has performed with or written for Academy Award winners, Emmy winners and huge box office entertainers such as Russell Crowe, Sylvester Stallone, Chelsea Handler, Ryan Gosling, James Cann, Will Ferrell - all of these can be verified through his credits on IMDB[[User:Thelinzla|Thelinzla]] ([[User talk:Thelinzla|talk]]) 23:12, 30 January 2017 (UTC)

2966 |caption = North American A cover artwork featuring [[Sylvester Stallone]] as [[John Rambo]]



```
2967 Director [[Ron Howard]] sees Kevin's play and casts him in a war movie.&lt;ref name=Ep17/&gt; Kevin has trouble with his lines when he cannot focus after Kate (visiting the set) talks with costar [[Sylvester Stallone]] about her dad, and then Stallone talks to Kevin about his dad. Kevin also re-injures his knee (an old football injury) during an explosion scene on the movie set.&lt;ref name=Ep21&gt;{{Cite episode|title=Déjà Vu|series=This is Us|network=NBC|airdate=October 10, 2017|season=2|number=3}}&lt;/ref&gt;
2968 * [[Debra Jo Rupp]] as Linda: A social worker who worked with Randall and Beth Pearson when they fostered Deja.&lt;ref name=&quot;LauriaRupp&quot;&gt;{{cite web|url=https://deadline.com/2017/08/this-is-us-tca-1202141576|title='This Is Us' Season 2 Details: EP Dan Fogelman Talks; Sylvester Stallone To Guest Star - TCA|last=D'Alessandro|first=Anthony|website=Deadline.com|publisher=Penske Business Media, LLC.|date=August 3, 2017|accessdate=September 4, 2017|url-status=live|archiveurl=https://web.archive.org/web/20170807130215/http://deadline.com/2017/08/this-is-us-tca-1202141576|archivedate=August 7, 2017}}&lt;/ref&gt;
2969 * [[Sylvester Stallone]] as himself: Actor in a movie Kevin was cast in.&lt;ref name=&quot;Stallone&quot;&gt;{{cite web|url=https://variety.com/2017/tv/news/this-is-us-season-2-sylvester-stallone-1202515046|title='This Is Us' Season 2 Adds Sylvester Stallone in Guest Starring Role|last=Otterson|first=Joe|work=[[Variety (magazine)|Variety]]|date=August 3, 2017|accessdate=September 4, 2017|url-status=live|archiveurl=https://web.archive.org/web/20170808232141/http://variety.com/2017/tv/news/this-is-us-season-2-sylvester-stallone-1202515046|archivedate=August 8, 2017}}&lt;/ref&gt;
2970 He made his feature film debut in ''Jara'' (1999), in which he starred alongside the also debuting [[Olivia Molina (actress)|Olivia Molina]] and seasoned actress [[Ángela Molina]].&lt;ref name=&quot;elpais&quot;&gt;{{Cite web|url=https://elpais.com/diario/1999/11/22/andalucia/943226538_850215.html|website=[[El País]]|first=Luis Eduardo|last=Siles|title=Ángela Molina y su hija Olivia brillan en Huelva con la película &quot;Jara&quot;|date=22 November 1999}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.cadena100.es/programas/de-sabado-con-christian-galvez/noticias/increible-conversacion-entre-sergio-peris-mencheta-sylvester-stallone-20210630_1373367|website=Cadena100|title=La increíble conversación entre Sergio Peris-Mencheta y Sylvester Stallone|date=30 June 2021}}&lt;/ref&gt; He portrayed Tato, a man who finds a mysterious and beautiful woman (Jara) living alone in the forest since she was a child, starting a murky relationship.&lt;ref&gt;{{Cite web|url=https://www.fotogramas.es/peliculas-criticas/a4049/jara/|website=[[Fotogramas]]|title=Jara|date=29 May 2008}}&lt;/ref&gt;
2971     <redirect title="Sylvester Stallone" />
2972       <comment>[[WP:AES|←]]Redirected page to [[Sylvester Stallone#Boxing promoter]]</comment>
2973       <text bytes="48" sha1="01u94kp8kp9ftfvnqamtvnl8sa6sofz" xml:space="preserve">#REDIRECT [[Sylvester Stallone#Boxing promoter]]</text>
2974 |[[Sylvester Stallone]] (in [[Rocky]])
2975 The '''first season''' of the American  reality competition series ''[[Ultimate Beastmaster]]'' premiered exclusively via [[Netflix]]'s web streaming service on February 24, 2017. The show consists of 10 Beastmaster episodes which were released simultaneously on Netflix worldwide.&lt;ref&gt;{{cite web|url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|title=Netflix Orders Global Competition Series 'Ultimate Beastmaster' from Sylvester Stallone, Dave Broome|last=Prudon|first=Laura|work=Variety|date=May 9, 2016|access-date=October 5, 2016}}&lt;/ref&gt; The show was filmed in [[Santa Clarita, California]], over the course of eight nights.&lt;ref&gt;[https://www.hollywoodreporter.com/news/how-netflixs-ultimate-beastmaster-will-change-global-reality-tv-976142 How Netflix's 'Ultimate Beastmaster' Will Change Global Reality TV], ''The Hollywood Reporter''&lt;/ref&gt;
2976 In addition to [[Sylvester Stallone]] hosting the series, each country has their own set of two hosts/commentators for the competition. They are as follows:
2977 The show has six country-specific versions. These have separate hosts, and languages, with two competitors from each country competing in each of the first nine episodes of the series. The countries are the [[United States|U.S.]], [[Brazil]], [[South Korea]], [[Mexico]], [[Germany]], and [[Japan]].&lt;ref name=details&gt;{{cite web|url=https://deadline.com/2016/05/netflix-ultimate-beastmaster-sylvester-stallone-competition-reality-series-1201751580/|title=Netflix Orders 'Ultimate Beastmaster' Competition Reality Series With Sylvester Stallone|last=Andreeva|first=Nellie|work=Deadline|date=May 9, 2016|access-date=October 5, 2016}}&lt;/ref&gt;
2978 The '''second season''' of the American reality competition series ''[[Ultimate Beastmaster]]'' premiered exclusively via [[Netflix]]'s web streaming service on December 15, 2017. The show consists of 10 Beastmaster episodes which were released simultaneously on Netflix worldwide.&lt;ref&gt;{{cite web|url=https://variety.com/2016/digital/news/ultimate-beastmaster-netflix-competition-series-sylvester-stallone-dave-broome-1201769095/|title=Netflix Orders Global Competition Series 'Ultimate Beastmaster' from Sylvester Stallone, Dave Broome|last=Prudon|first=Laura|work=Variety|date=May 9, 2016|accessdate=October 5, 2016}}&lt;/ref&gt; The show was filmed in [[Santa Clarita, California]], over the course of eight nights.&lt;ref&gt;[http://www.hollywoodreporter.com/news/how-netflixs-ultimate-beastmaster-will-change-global-reality-tv-976142 How Netflix's 'Ultimate Beastmaster' Will Change Global Reality TV]&lt;/ref&gt;
2979 | ''[[The Family Stallone]]''&lt;ref&gt;{{cite web|title=Sylvester Stallone's Family To Star In Docuseries For Paramount+, Front And Center In Super Bowl Ad|url=https://deadline.com/2023/02/sylvester-stallone-family-docuseries-for-paramount-1235247510/|date=February 2, 2023|access-date=February 16, 2023}}&lt;/ref&gt; || Coming 2023 || [[Paramount+]] || co-production with [[MTV Entertainment Studios]]
2980 | Dolly Parton and [[Sylvester Stallone]]
2981 | [[Sylvester Stallone]] with Dolly Parton
2982 | Dolly Parton and [[Sylvester Stallone]]
2983 | Dolly Parton and [[Sylvester Stallone]]
2984 How do you make a box like the one on [[Sylvester Stallone]] page..like where its got his picture that kind of box.
2985 Other Latin Lovers of world cinema, partially or through their whole careers, include [[Warner Baxter]], [[Iván Petrovich]], [[Pierre Blanchard]], [[George Raft]], [[Cesar Romero]], [[Fernando Lamas]], [[Ricardo Montalbán]], [[Sal Mineo]], [[John Gavin]], [[George Hamilton (actor)|George Hamilton]],&lt;ref name=&quot;FE&quot;/&gt;&lt;ref name=culture/&gt;&lt;ref name=ndn&gt;{{cite news | url = http://www.nydailynews.com/entertainment/tv-movies/hollywood-latin-lovers-long-lurid-history-article-1.1612214 | author = Robert Dominguez | title = Hollywood's Latin Lovers have a long and lurid history | date = February 12, 2014 | newspaper = [[New York Daily News]]}}&lt;/ref&gt; [[Jean-Paul Belmondo]], [[Alain Delon]], [[Al Pacino]], [[Sylvester Stallone]], [[John Travolta]], [[Andy García]],&lt;ref name=thomas/&gt; [[Olivier Martinez]]&lt;ref&gt;{{cite magazine | url = https://www.vogue.it/en/uomo-vogue/look-of-the-day/2011/05/olivier-martinez?refresh_ce= | author = Camilla Piccitto | title = Olivier Martinez - The cool look of a modern Casanova | date = May 16, 2011 | magazine = [[Vogue (magazine)|Vogue]]}}&lt;/ref&gt; and [[Vincent Perez]].&lt;ref name=orlando/&gt;
2986 | [[Sylvester Stallone]]
2987 Lionsgate has a number of additional projects in various stages of development, set within the same [[Shared universe|fictional continuity]] as the ''John Wick'' films. In March 2023, producer Erica Lee stated that in addition to previous identified projects, there's &quot;... lots of other things. But we're developing a lot of stuff and having a lot of discussions with a lot of writers and brand management and ''Wick''... is my utmost priority.&quot;&lt;ref name=&quot;JW.franchise_Collider&quot; /&gt; Stahelski has expressed interest in seeing additional actors join the franchise, such as: [[Cillian Murphy]], [[Jet Li]], [[Jackie Chan]], [[Bob Odenkirk]], [[Jason Statham]], [[Sylvester Stallone]], [[Clint Eastwood]], [[Jason Momoa]], [[Matt Damon]], [[Chris Hemsworth]], [[Colin Farrell]], [[Charlize Theron]], [[Michelle Yeoh]], [[Peter Dinklage]], [[Jurnee Smollett]], [[Robert Downey Jr.]], [[Sean Bean]], [[Jeff Wincott]], [[Jeff Speakman]], [[Tyrese Gibson]], [[Ludacris]], [[Jamie Foxx]] and [[Brad Pitt]].&lt;ref name=&quot;Future_CB&quot;&gt;{{cite web|url=https://comicbook.com/movies/news/john-wick-5-director-teases-sequel-robert-downey-jr/|work=ComicBook.com|title=John Wick Director Reveals Which Actors He Wants for Sequel Film|author=Barnhardt, Adam|date=April 18, 2023|access-date=April 18, 2023}}&lt;/ref&gt;&lt;ref name=&quot;Future_Dexerto&quot;&gt;{{cite web|url=https://www.dexerto.com/tv-movies/new-mystery-john-wick-spinoff-in-the-works-after-chapter-4-2097785/|work=Dexerto|title=John Wick director says he'd &quot;love&quot; Jet Li &amp; Jackie Chan to star in next sequel|author=Gwilliam, Michael|date=March 27, 2023|access-date=March 27, 2023}}&lt;/ref&gt;&lt;ref name=&quot;Chapter5_SR&quot; /&gt; Lee stated that with various projects in development, the studio may develop some prequel installments as well.&lt;ref name=&quot;Future_SF&quot;&gt;{{cite web|url=https://www.slashfilm.com/1239461/john-wick-producer-erica-lee-on-more-spin-offs-and-what-didnt-make-it-into-chapter-4-exclusive-interview/|work=Slash Film|title=John Wick Producer Erica Lee On More Spin-Offs And What Didn't Make It Into Chapter 4 [Exclusive Interview]|author=Armstrong, Vanessa|date=March 27, 2023|access-date=March 27, 2023}}&lt;/ref&gt; That same month, [[Natalia Tena]] expressed interest in reprising her role as Wick's adoptive cousin/surrogate sister Katia, in future installments of the franchise.&lt;ref name=&quot;Katia_CBM&quot;&gt;{{cite web|url=https://comicbookmovie.com/action/john-wick/john-wick-chapter-4-actress-natalia-tena-on-playing-johns-spoiler-an-alternate-ending-more-exclusive-a201311|work=Comic Book Movie|title=JOHN WICK: CHAPTER 4 Actress Natalia Tena On Playing John's SPOILER, An Alternate Ending, &amp; More (Exclusive)|author=Patel, Rohan|date=March 29, 2023|access-date=March 30, 2023}}&lt;/ref&gt; In November 2023, Stahelski confirmed that multiple projects were in development, while acknowledging that there are plans to explore the various criminal organization establishments within the continuity; including Italian, French, Japanese, Chinese, Middle-Eastern, Indonesian, and Russian cultures.&lt;ref name=&quot;Future_SR&quot;&gt;{{cite web|url=https://screenrant.com/john-wick-tv-show-characters-return-franchise-director-tease/|work=Screen Rant|title=John Wick Franchise Director Teases Which Characters Could Return For New TV Show: &quot;Worlds Are Going To Be Explored&quot;|author=Hermanns, Grant|date=November 6, 2023|access-date=November 6, 2023}}&lt;/ref&gt; Later following the resolutions to the 2023 writers and actors strikes, Lionsgate officially announced that work on various spin-off projects had commenced.&lt;ref name=&quot;Future_CB.com&quot; /&gt;
2988 | ''[[Escape Plan 2: Hades]]'' || [[Steven C. Miller]] || [[Sylvester Stallone]], [[Huang Xiaoming]], [[Dave Bautista]] || Action / Thriller || Mainland-United States co-production || style=&quot;text-align:center;&quot;| &lt;ref&gt;{{cite web |url=https://movie.douban.com/subject/26905469 |title=金蝉脱壳 2 Escape Plan 2: Hades |accessdate=2018-06-25}}&lt;/ref&gt;
2989 *2022 ''[[Tulsa King]]'' starring [[Sylvester Stalone]] visually features the book and his character is said to have read it during his incarceration.&lt;ref&gt;{{cite news |title=Dwight's Books In Tulsa King Secretly Reveal 5 Details About Him |url=https://screenrant.com/tulsa-king-dwight-books-prison-meaning-reveals/ |agency=ScreenRant}}&lt;/ref&gt;
2990 * [[Sylvester Stallone]]
2991 '''''Escape Plan 2: Hades''''' is a 2018 American [[direct-to-video]] [[prison film|prison]] [[action thriller film]] directed by [[Steven C. Miller]]. It is the sequel to the 2013 film ''[[Escape Plan (film)|Escape Plan]]'', and the second installment in the [[Escape Plan (film series)|''Escape Plan'' film series]]. It stars [[Sylvester Stallone]] and [[50 Cent|Curtis &quot;50 Cent&quot; Jackson]] reprising their roles from the first film, with [[Dave Bautista]], [[Huang Xiaoming]], [[Jaime King]], [[Jesse Metcalfe]], [[Titus Welliver]], and [[Wes Chatham]] joining the cast. ''Escape Plan 2: Hades'' was released [[straight-to-DVD]] in the United States but received theatrical releases in such countries as Russia on June 28, 2018, and in China on June 29, 2018. The film received negative reviews from critics; it grossed $17.6 million&lt;ref name=&quot;BOM&quot;&gt;{{cite web|title=Escape Plan 2: Hades (2018)|url=https://www.boxofficemojo.com/title/tt6513656/|website=[[Box Office Mojo]]|access-date=June 29, 2021}}&lt;/ref&gt; in some theaters and $4.2 million&lt;ref&gt;{{cite web|title=''Escape Plan 2: Hades'' (2018)|url=https://www.the-numbers.com/movie/Escape-Plan-2-Hades-(2018)|work=[[The Numbers (website)|The Numbers]]|access-date=June 29, 2021}}&lt;/ref&gt; in domestic home market against a production budget of $20 million.
2992 * [[Sylvester Stallone]] as Ray Breslin, the head of Breslin Security
2993 Development was first announced in October 2016, with [[Sylvester Stallone]] returning to star.&lt;ref&gt;{{cite magazine|url=https://collider.com/escape-plan-2-sylvester-stallone/|title='Escape Plan 2': Sylvester Stallone Planning Another Breakout|date=31 October 2016|magazine=[[Complex (magazine)|Complex Media]]|first=Nick|last=Romano}}&lt;/ref&gt; It was announced in February 2017 that [[Steven C. Miller]] would helm the film, and that there was potential for [[Arnold Schwarzenegger]] to reprise his role from the first.&lt;ref&gt;{{cite web|url=https://www.comingsoon.net/movies/news/812145-escape-plan-2-director-confirmed-as-steven-c-miller#/slide/1|title=Escape Plan 2 Director Confirmed as Steven C. Miller|date=8 February 2017|publisher=[[CraveOnline Media]]|work=ComingSoon.net|access-date=June 29, 2021|first=Max|last=Evry}}&lt;/ref&gt; Chinese film production company Leomus Pictures was set as a co-financer for the
```



film.&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/news/sylvester-stallones-escape-plan-2-set-as-china-production-970890|title=Sylvester Stallone's 'Escape Plan 2' Set as China Co-Production|publisher=[[Prometheus Global Media]]|work=[[The Hollywood Reporter]]|access-date=June 29, 2021|first=Patrick|last=Brzeski|date=1 February 2017}}&lt;/ref&gt; In March 2017, [[Dave Bautista]], [[Jaime King]] and [[50 Cent]] were added to the cast,&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/heat-vision/dave-bautista-joining-sylvester-stallone-escape-plan-2-985073|title=Dave Bautista Joining Sylvester Stallone in 'Escape Plan 2' (Exclusive)|publisher=[[Prometheus Global Media]]|work=[[The Hollywood Reporter]]|access-date=June 29, 2021|first=Borys|last=Kit|date=10 March 2017}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://deadline.com/2017/03/jaime-king-curtis-50-cent-jackson-escape-plan-2-sylvester-stallone-1202041564/|title=Jaime King &amp; Curtis '50 Cent' Jackson Break Into 'Escape Plan 2'|first=Erik|last=Pedersen|date=11 March 2017|publisher=[[Penske Business Media]]|work=[[Deadline Hollywood]]|access-date=June 29, 2021}}&lt;/ref&gt; with the additions of [[Jesse Metcalfe]], [[Pete Wentz]] and [[Wes Chatham]] later in the month.&lt;ref&gt;{{cite web|url=http://www.tracking-board.com/jesse-metcalfe-pete-wentz-join-sylvester-stallone-in-escape-plan-2/|title=Jesse Metcalfe, Pete Wentz Join Sylvester Stallone in &quot;Escape Plan 2&quot;|date=21 March 2017|work=The Tracking Board|access-date=June 29, 2021|first=Jeff|last=Sneider}}&lt;/ref&gt;

2994 David Ehrlich of [[IndieWire]] gave the film a &quot;D-&quot; and wrote: &quot;''Hades'' might boast some decent star power, but there isn't a celebrity in the world who could save this bargain bin nonsense from feeling like a bootlegged ripoff of its own franchise. Not even Arnold Schwarzenegger, who faced off against Stallone in the first one, could have rescued this from abject boredom.&quot;&lt;ref&gt;{{cite web|last=Ehrlich|first=David|date=29 June 2018|title='Escape Plan 2: Hades' Review: Sylvester Stallone and Dave Bautista Are Trapped in Movie Hell|url=https://www.indiewire.com/2018/06/escape-plan-2-hades-review-sylvester-stallone-dave-bautista-1201979919/|publisher=[[Penske Business Media]]|work=[[IndieWire]]|access-date=June 29, 2021}}&lt;/ref&gt;

2995 A third film entered the early stages of development even before ''Escape Plan 2: Hades'' wrapped filming. Stallone was again signed on to reprise his role as Ray Breslin.&lt;ref&gt;{{cite news|url=https://www.comingsoon.net/movies/news/837111-escape-plan-3-set-with-sylvester-stallone-to-return#/slide/1|title=Escape Plan: The Extractors Set with Sylvester Stallone to Return|date=12 April 2017|publisher=CraveOnline Media|work=ComingSoon.net|access-date=June 29, 2021|first=Max|last=Evry}}&lt;/ref&gt; Dave Bautista also signed on to reprise his role in ''[[Escape Plan: The Extractors]]'' and filming began in September 2017. ''Escape Plan: The Extractors'' had a smaller production budget, but received more positive reviews compared to the second film.

2996 | {{sort| Stallone | [[Sylvester Stallone]]}} || M || 1946 || ~ || {{Age| 1946 |7|6}} || 6712 || 1984 || 38 || {{nominated|Nom}}

2997 '''''Another WolfCop''''' is a 2017 Canadian [[Superhero film|superhero]] [[horror film|horror]] [[comedy film]] written and directed by [[Lowell Dean]] and is the sequel to the 2014 film ''[[WolfCop]]''. It was released in theatres December 1, 2017.&lt;ref&gt;{{Cite web | url=http://bloody-disgusting.com/movie/3470032/another-wolfcop-poster-prepares-night-chaos-exclusive/ | title='Another WolfCop' Poster Prepares for a Night of Chaos [Exclusive] | last=Miska | first=Brad | date=2017-11-15 | website=[[Bloody Disgusting]] | access-date=2017-11-15}}&lt;/ref&gt; The poster was modeled on the 1986 [[Sylvester Stallone]] film ''[[Cobra (1986 film)|Cobra]]''.&lt;ref name=&quot;poster&quot;&gt;{{cite web |date=3 February 2019 |title=Another Wolfcop B |url=https://thedudedesigns.com/another-wolfcop-b/?v=79cba1185463 |website=The Dude Designs |url-status=live |archive-url=https://web.archive.org/web/20200919212126/https://thedudedesigns.com/another-wolfcop-b/?v=79cba1185463 |archive-date=2020-09-19 }}&lt;/ref&gt;&lt;ref&gt;{{cite web |date=26 September 2016 |author=B. Alan Orange |title='WolfCop 2' Poster Spoofs Sylvester Stallone's 'Cobra' |url=https://movieweb.com/wolfcop-2-poster-cobra-spoof/ |website=Movieweb }}&lt;/ref&gt;

2998 The American judiciary has followed both these tests. There have also been a number of instances where the Court has used both the tests in its analysis. For instance, in ''[[Anderson v. Stallone|Anderson v Stallone]]'',&lt;ref&gt;11 USPQ2D 1161 (C.D. Cal. 1989).&lt;/ref&gt; it was held that the character [[Rocky Balboa|Rocky]] played by movie star [[Sylvester Stallone]] in the eponymous movie franchise had come to be identified with certain physical traits and mannerisms that qualified as a well delineated character. Further, this 'highly developed' character was held to constitute the story being told. Similarly, the character of [[James Bond (literary character)|James Bond]] was awarded copyright protection on the grounds of being both sufficiently delineated as well as central to the story being told.&lt;ref&gt;''Metro-Goldwyn-Mayer v. American Honda'', 900 F. Supp. 1287 (C.D. Cal. 1995).&lt;/ref&gt; The fact that the character had been played by multiple actors was not held to be relevant since the characteristics associated with Bond had remained consistent across sixteen films.

2999 In 2017 the Stallone Sisters, Sistine, Sophia, and Scarlet, the three daughters of movie star [[Sylvester Stallone]], were featured in the #MeetTheStallones campaign promoting a new capsule collection for TDE.&lt;ref&gt;{{cite web|url=http://www.vogue.com.au/fashion/news/the+daily+edited+stallone+sisters+new+campaign,42526/|title=The Stallone sisters are the new faces of The Daily Edited|date=18 April 2017|publisher=}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=http://www.vanityfair.com/style/2017/04/sylvester-stallone-daughters-fashion-campaign/|title=Sylvester Stallone Calls His Daughters' New Fashion Campaign &quot;Volcanic&quot;|first=Erika|last=Harwood|website=[[Vanity Fair (magazine)|Vanity Fair]] |date=18 April 2017 |publisher=}}&lt;/ref&gt;

3000 *'''Bicep''' (voiced by Roly Gutierrez (English)&lt;ref name=&quot;merged2&quot;&gt;{{cite tweet |author=Caja Cartoon |user=CajaCN |number=1586760380923224067 |date=October 30, 2022 |title=Créditos del doblaje en Inglés del episodio. Las voces de Dolencia, el Dr. Flex, G-Lo y Bicep reveladas. DOBLAJE AÚN NO DISPONIBLE. https://t.co/8Nw07OQuBa |language=es |access-date=November 29, 2022 | archive-url=https://web.archive.org/web/20221030174743/https://twitter.com/cajacn/status/1586760380923224067 |archive-date=October 30, 2022 |url-status=live}}&lt;/ref&gt; and Daniel del Robe (Spanish)), a boxer superhero who resembles [[Sylvester Stallone]]'s [[Rocky Balboa]] character from the Rocky film series, whom has superhuman strength.

3001 |[[Sylvester Stallone]]

3002 |&lt;ref&gt;{{Cite web |last=Setoodeh |first=Ramin |date=2016-01-09 |title=Sylvester Stallone on Donald Trump, Republicans and Running for Office |url=https://variety.com/2016/film/news/sylvester-stallone-donald-trump-1201675595/ |access-date=2023-11-29 |website=Variety |language=en-US}}&lt;/ref&gt;

3003 | A film about [[Robert De Niro]] (Billy &quot;The Kid&quot; McDonnen) and [[Sylvester Stallone]] (Henry &quot;Razor&quot; Sharp) as boxing rivals. In the beginning of the film, both actors are de-aged to show them fighting each other in their youth/prime. The rest of the film is set several decades later.

3004 | A de-aged [[Sylvester Stallone]] appears in a flashback scene.

3005 '''Marco de la O''' is a Mexican actor, best known for the lead role of [[Joaquín &quot;El Chapo&quot; Guzmán]] in the [[Netflix]] and [[Univision]] television series ''[[El Chapo (TV series)|El Chapo]]''.&lt;ref&gt;{{cite news |first=Veronica |last=Villafañe |url=https://www.forbes.com/sites/veronicavillafane/2017/03/24/marco-de-la-o-lands-el-chapo-lead-in-univision-netflix-series-set-for-april-debut/#2e7a9b6411bf |title=Marco De La O Lands 'El Chapo' Lead In Univision-Netflix Series Set For April Debut |magazine=[[Forbes]] |date=March 24, 2017 |accessdate=28 June 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite news |first=James |last=Donaghy |url=https://www.theguardian.com/tv-and-radio/2017/jun/27/el-chapo-netflix-gruesome-gripping-answer-to-the-wire-mexico |title=El Chapo: Netflix's gruesome, gripping answer to The Wire |newspaper=[[The Guardian]] |location=[[London]] |date=27 June 2017 |accessdate=28 June 2023}}&lt;/ref&gt; He starred opposite [[Sylvester Stallone]] in ''[[Rambo: Last Blood]]'' in 2019. He starred in a lead role opposite [[Paulina Dávila]] and [[Mauricio Ochmann]] in the limited television series ''R'', a joint venture between [[Viacom (2005-present)|Viacom International]] and Spanish-language distributor Clarovideo in mid-2020. He is set to star in the second season of ''[[Falsa Identidad]]'' opposite [[Camila Sodi]], [[Luis Ernesto Franco]], [[Eduardo Yañez]], and [[Sonya Smith]] in late 2020.

3006 Not notable as it is a proposed television show that was meant to be made but never was. Promotional videos were made, but no episodes ever aired on TV, and no episodes as far as I can tell were ever made. Majority of references are related to Sylvester Stallone and the lawsuit mentioned in article. Two of the other sources are self-published by the proposed TV show and/or the creator. Potential COI by one of the editors ([[User:Rob Fletcher]]), but his edits were reverted.

3007 Nominating for deletion, information could potential be merged/added into either [[Sylvester Stallone]] or [[Strong (TV series)]] [[User:WikiVirusC|WikiVirusC]] ([[User talk:WikiVirusC|talk]]) 19:19, 30 June 2017 (UTC)

3008 After graduating, Rossall traveled to [[Europe]] and continued to surf. Upon his return, he had met stuntman [[Terry Leonard]] while working as a ski patrolman at [[Mammoth Mountain]]. Rossall was accepted to the [[Naval aviation|Naval Aviation Program]] in [[Pensacola, Florida]] for [[United States Marine Corps|Marine Corps]] boot camp and was [[Flight training|learning to be a pilot]] when the [[Vietnam War]] ended. His [[aeronautics]] training helped prepare him for a career in the [[film industry]].&lt;ref name=&quot;:2&quot; /&gt; Despite being credited among stars like [[Arnold Schwarzenegger]] and [[Sylvester Stallone]], he has remained out of the news for most of his career.&lt;ref name=&quot;:1&quot;&gt;{{Cite web |last=Kantor |first=Jonathan H. |date=2022-09-28 | title=The Only Actors Still Alive From The Cast Of Apocalypse Now |url=https://www.looper.com/1030381/the-only-actors-still-alive-from-the-cast-of-apocalypse-now/ |access-date=2022-11-01 | website=[[Looper (website)|Looper]] |language=en-US}}&lt;/ref&gt; Rossall resides in [[Carlsbad, California]].&lt;ref name=&quot;:0&quot;&gt;{{Cite web |last=Elling |first=Regina |date=2022-10-18 | title=Sunset Cliffs adds touch of beauty to apocalyptic San Diego-made film 'Friend of the World' |url=https://www.pointloma-obmonthly.com/news/story/2022-10-18/sunset-cliffs-adds-touch-of-beauty-to-apocalyptic-san-diego-made-film-friend-of-the-world |access-date=2022-11-01 |website=Point Loma-OB Monthly |language=en-US}}&lt;/ref&gt;

3009 In the movie &quot;Demolition Man,&quot; Sylvester Stallone's character inadvertently orders a hamburger made of rat meat.

3010 |373||July 8, 1982||[[Sylvester Stallone]]||

3011 The album incorporates new instrumentation for the duo: the use of a [[drum machine]] and &quot;a more expansive guitar palette&quot;.&lt;ref name=&quot;Rolling Stone&quot;&gt;{{cite web|url=https://rollingstoneaus.com/reviews/post/angus-and-julia-stone-snow/7062|title=Rolling Stone Australia – Angus &amp; Julia Stone, 'Snow'|work=[[Rolling Stone Australia]]|last=Levin|first=Darren|date=14 September 2017|accessdate=16 September 2017}}&lt;/ref&gt; It also incorporates the use of a $150 organ the pair bought from an advertisement on the opening track, &quot;Snow&quot;.&lt;ref name=&quot;News.com.au&quot;/&gt;&lt;ref&gt;{{cite web|url=http://www.gettothefront.co.uk/angus-julia-stone-snow/|title=Angus &amp; Julia Stone. 4th Album SNOW – teaser|work=Get to the Front|last=7 July 2017|accessdate=16 September 2017}}&lt;/ref&gt; The track &quot;Sylvester Stallone&quot; includes lyrics referring to a lover who slurs their speech when drunk,&lt;ref name=&quot;Telegraph&quot;/&gt; and &quot;Sleep Alone&quot; was created after a &quot;'[[Speed dating|speed-date]]' songwriting session&quot; with an unnamed DJ, where the duo abandoned the programmed beat but kept the lyrics.&lt;ref name=&quot;Telegraph&quot;/&gt;

3012 | title12 = Sylvester Stallone

3013 * [[Sylvester Stallone]] as himself, an actor in Kevin's movie.&lt;ref name=&quot;Stallone&quot;&gt;{{cite web |url=https://variety.com/2017/tv/news/this-is-us-season-2-sylvester-stallone-1202515046/ |title='This Is Us' Season 2 Adds Sylvester Stallone in Guest Starring Role |last=Otterson |first=Joe |work=[[Variety (magazine)|Variety]] |date=August 3, 2017 |access-date=September 4, 2017 |url-status=live |archive-url=https://web.archive.org/web/20170808232141/http://variety.com/2017/tv/news/this-is-us-season-2-sylvester-stallone-1202515046/ |archive-date=August 8, 2017 |df=mdy-all }}&lt;/ref&gt;

3014 * [[Dan Lauria]] as Mr. Damon, Toby's father.&lt;ref name=&quot;LauriaRupp&quot;&gt;{{cite web|url=https://deadline.com/2017/08/this-is-us-tca-1202141576/|title='This Is Us' Season 2 Details: EP Dan Fogelman Talks; Sylvester Stallone To Guest Star – TCA|last=D'Alessandro|first=Anthony|website=[[Deadline Hollywood]]|date=August 3, 2017|access-date=September 4, 2017|url-status=live|archive-url=https://web.archive.org/web/20170807130215/http://deadline.com/2017/08/this-is-us-tca-1202141576/|archive-date=August 7, 2017|df=mdy-all}}&lt;/ref&gt;&lt;ref name=&quot;LauriaMalick&quot;&gt;{{cite web|url=http://www.eonline.com/news/920441/toby-s-parents-on-this-is-us-are-tv-legends|title=Toby's Parents on This Is Us Are TV Legends|last=Harnick|first=Chris|work=[[E!]]|date=March 13, 2018|access-date=March 14, 2018|archive-url=https://web.archive.org/web/20180314011555/http://www.eonline.com/news/920441/toby-s-parents-on-this-is-us-are-tv-legends|archive-date=March 14, 2018|url-status=live}}&lt;/ref&gt;

3015 |ShortSummary   = In flashbacks, when the kids are teenagers, Jack talks about his efforts to rehabilitate from alcoholism. Randall tries to find his birth parents through an ad, and is comforted by Kate and Kevin after a woman falsely claims to be his birth mother. In the present, Kate attends Kevin's movie set. Excited that [[Sylvester Stallone]] is his co-star, Kate bonds with Stallone and

tells him what his films meant to Jack. After talking with Stallone, Kevin's memories of his dad surface, affecting his performance in a scene. Randall learns that a young girl, Deja, is being brought to his home. Later, when Deja yells at Beth for &quot;stealing&quot; her mom's cigarettes, Randall defends his wife, and Deja shrinks to defend herself from apparent aggression. Randall talks to Deja about the joy he found in having Jack and Rebecca, as well as William. Kevin takes pain medication after his old football knee injury is reinjured during the filming of a battle scene.

3016 In August 2017, [[Sylvester Stallone]] and [[Debra Jo Rupp]] were cast in guest starring roles.&lt;ref name=&quot;Stallone&quot; /&gt;&lt;ref name=&quot;Rupp&quot; /&gt;

3017 * [[Sylvester Stallone]]

3018 | budget   = $3.6 million&lt;ref name=&quot;budget and success&quot;&gt;{{cite web|url=https://www.cinemablend.com/news/2476477/sylvester-stallone-is-so-proud-of-escape-plan-3-success-since-it-was-made-in-just-17-days|title=Sylvester Stallone Is 'So Proud' Of Escape Plan 3 Success, Since It Was Made In Just 17 Days|date=2019-07-14|publisher=CinemaBlend|access-date=2019-11-17}}&lt;/ref&gt;

3019 '''''Escape Plan: The Extractors''''' is a 2019 American [[direct-to-video]] [[prison film|prison]] [[action thriller film]] directed and co-written by [[John Herzfeld]], and a sequel to ''[[Escape Plan (film)|Escape Plan]]'' (2013) and ''[[Escape Plan 2: Hades]]'' (2018). It is the third installment in the [[Escape Plan (film series)|''Escape Plan'' film series]]. The film features [[Sylvester Stallone]], [[50 Cent]], [[Dave Bautista]] and [[Jaime King]] reprising their roles from the previous films with [[Zhang Jin (actor)|Max Zhang]], [[Harry Shum Jr.|Harry Shum, Jr.]], [[Malese Jow]] and [[Devon Sawa]] joining the cast. The film was released [[straight-to-DVD]] in the United States on July 2, 2019, but received theatrical releases in such countries as Russia, Italy, Australia, Turkey, Egypt and Portugal.&lt;ref name=&quot;NUM&quot;&gt;{{cite web |url=https://www.the-numbers.com/movie/Escape-Plan-The-Extractors#tab=summary|title=Escape Plan: The Extractors|website=The Numbers |publisher=Nash Information Services}}&lt;/ref&gt;

3020 * [[Sylvester Stallone]] as Ray Breslin, founder of Breslin Security

3021 Development was first announced in April 2017, when Stallone said while filming ''Escape Plan 2: Hades'' that a third installment in this franchise was on its way. In August 2017, casting was underway for extras and speaking roles. The producers aimed for actors with [[Mixed martial arts|MMA]] experience, as there would be potential fight scenes with Stallone and Bautista.&lt;ref&gt;{{cite web |date= Aug 17, 2017 |author= Emily Mills |title= Hollywood calling: 'Escape Plan 3' seeks extras |url= http://www.mansfieldnewsjournal.com/story/news/local/2017/08/17/hollywood-calling-escape-plan-3-seeks-extras/575113001/ |work= [[Mansfield News Journal]] |publisher= Gannett }}&lt;/ref&gt; Filming began on September 18, 2017, and wrapped on October 13, 2017.&lt;ref&gt;{{cite web | title= Sylvester Stallone Shares Videos from the Escape Plan: The Extractors Set|date=19 September 2017 |url= http://www.comingsoon.net/movies/news/889001-sylvester-stallone-shares-videos-from-the-escape-plan-3-set |publisher= ComingSoon.net }}&lt;/ref&gt; The film was originally titled ''Escape Plan 3: Devil's Station'', but by October 2018 the title had been changed to ''Escape Plan: The Extractors''.&lt;ref&gt;{{cite web | title=Escape Plan the Extractors| website=Lionsgate.com| url=https://www.lionsgate.com/movies/escape-plan-the-extractors | access-date=11 June 2023}}&lt;/ref&gt;

3022 Noel Murray of the [[Los Angeles Times]] said that &quot;while this film is better than its predecessor&quot; it lacks the spark of the original and &quot;is, ultimately, just another [[video on demand|VOD]] actioner.&quot;&lt;ref&gt;{{cite news |date=3 July 2019 |author=Noel Murray |title=Reviews: Sylvester Stallone and Dave Bautista in 'Escape Plan: The Extractors'; and more |url=https://www.latimes.com/entertainment/movies/la-et-mn-capsule-review-string-20190703-story.html |newspaper=[[Los Angeles Times]] }}&lt;/ref&gt;

3023 | {{dts|2017|6|23|format=y}} || ''[[Animal Crackers (2017 film)|Animal Crackers]]'' || Talia || || &lt;ref&gt;{{cite magazine | url=http://insidemovies.ew.com/2014/11/06/ian-mckellen-danny-devito-sylvester-stallon-animal-crackers/ | title=Ian McKellen, Danny DeVito, and Sylvester Stallone sign on for 'Animal Crackers' | magazine=[[Entertainment Weekly]] | date=November 6, 2014 | last=Smith | first=C. Molly | url-status=live | archive-url=https://web.archive.org/web/20141229094929/http://insidemovies.ew.com/2014/11/06/ian-mckellen-danny-devito-sylvester-stallon-animal-crackers/ | archive-date=December 29, 2014 | df=mdy-all }}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.facebook.com/AnimalCrackersMovie/photos/a.231703060346991.1073741830.230270440490253/323680554482574/?type=1|title=Final SNEAK PEEK of the week...Talia will be played by the Inimitable (I looked it up... it means incapable of being imitated, matchless, irreplaceable) Tara Strong!...|publisher=[[Facebook]]|author=[[Scott Christian Sava]]|date=November 1, 2014|access-date=July 26, 2016|url-status=live|archive-url=https://web.archive.org/web/20170409141054/https://www.facebook.com/AnimalCrackersMovie/photos/a.231703060346991.1073741830.230270440490253/323680554482574/?type=1|archive-date=April 9, 2017}}&lt;/ref&gt;

3024 |&lt;ref&gt;{{cite web|last=Bechara|first=Diego Ramos|title=Sylvester Stallone-Produced Feature 'Lost on a Mountain In Maine' Slated for November Release by Blue Fox Entertainment (EXCLUSIVE)|url=https://variety.com/2024/film/news/lost-on-a-mountain-in-maine-blue-fox-entertainment-sylvester-stallone-1236085210/|website=[[Variety (magazine)|Variety]]|date=July 25, 2024|access-date=July 25, 2024}}&lt;/ref&gt;

3025   <title>File:Sylvester Stallone as Stakar Ogord.jpg</title>

3026   <comment>Favre1fan93 moved page [[File:Sylvester Stallone as Stakar Ogord in Guardians of the Galaxy Vol 2.jpg]] to [[File:Sylvester Stallone as Stakar Ogord.jpg]] without leaving a redirect: [[WP:FNC#6]]</comment>

3027 |Description = Sylvester Stallone as Stakar Ogord in the film ''[[Guardians of the Galaxy Vol. 2]]''.

3028 In 1990, it was reported that Hughes would direct [[Sylvester Stallone]] and [[John Candy]] in a comedy he had written titled ''Bartholomew vs. Neff'' for [[Carolco Pictures]].&lt;ref name=lat&gt;{{cite news|title=SHORT TAKES : Stallone in Line for Comedy Role|date=30 July 1990|newspaper=[[Los Angeles Times]]|url=https://www.latimes.com/archives/la-xpm-1990-07-30-ca-997-story.html|access-date=19 August 2018}}&lt;/ref&gt;&lt;ref name=crane&gt;{{cite book|url=https://books.google.com/books?id=ReptBgAAQBAJ&amp;q=bartholomew+vs+neff&amp;pg=PA252|last2=Fryer|first2=Christopher|last1=Crane|first1=Robert|title=Crane: Sex, Celebrity, and My Father's Unsolved Murder|year=2015|publisher=University Press of Kentucky|isbn=9780813160764}}page 252&lt;/ref&gt; The film was to have been about feuding neighbors.&lt;ref&gt;{{cite news|last=Carter|first=Bill|title=Him Alone|date=4 August 1991|newspaper=[[The New York Times]]|url=https://www.nytimes.com/1991/08/04/magazine/him-alone.html|access-date=19 August 2018}}&lt;/ref&gt; Hughes had planned to direct the film right after he finished ''[[Curly Sue]]'' (1991).&lt;ref name=evans/&gt;&lt;ref&gt;{{cite web|last=Evans|first=Bradford|title=The Lost Roles of John Candy|date=2 June 2011|publisher=[[Vulture.com]]|url=http://www.vulture.com/2011/06/the-lost-roles-of-john-candy.html|access-date=19 August 2018}}&lt;/ref&gt; According to the ''[[Los Angeles Times]]'', principal photography was scheduled to take place in the suburbs of Chicago during the summer of 1991.&lt;ref name=lat/&gt; The film was never made.&lt;ref name=evans/&gt;&lt;ref name=crane/&gt;

3029 Sherwood Oaks Film School's was originally founded as an alternative high school, dubbed Sherwood Oaks Experimental High School. Gary Shusett, brother of [[Alien (film franchise)|''Alien'' film franchise]] co-creator [[Ronald Shusett]],&lt;ref&gt;{{Cite news|last=Nelson|first=Valerie J.|date=2013-08-14|title=Gary Shusett dies at 72; co-founder of Sherwood Oaks Film School|language=en-US|work=Los Angeles Times|url=http://articles.latimes.com/2013/aug/14/local/la-me-gary-shusett-20130815|archive-url=https://web.archive.org/web/20130822114756/http://articles.latimes.com/2013/aug/14/local/la-me-gary-shusett-20130815|url-status=dead|archive-date=August 22, 2013|access-date=2017-09-10|issn=0458-3035}}&lt;/ref&gt; joined the school as headmaster in 1968. He renamed the school the Sherwood Oaks Experimental College in 1972, and began to offer classes to adults in a range of fields including batik and shipbuilding in bottles. In 1974, with the school on the verge of closing, Shusett contacted film industry celebrities and requested that they lecture at the school.&lt;ref name=&quot;:1&quot;&gt;{{Cite news|date=1977-10-10|title=It's Professor Sly Stallone, Among Other Volunteer Lecturers, at a Unique Hollywood 'College' - Vol. 8 No. 15|language=en-US|work=PEOPLE.com|url=http://people.com/archive/its-professor-sly-stallone-among-other-volunteer-lecturers-at-a-unique-hollywood-college-vol-8-no-15/|access-date=2017-09-10}}&lt;/ref&gt; The school came to offer classes on a broad range of filmmaking topics, with lectures from notable Hollywood names. The school changed its name to Sherwood Oaks College in 2010, and to Sherwood Oaks Film School in 2014.&lt;ref name=&quot;:2&quot;&gt;{{Cite web|title=Our History {{!}} Sherwood Oaks Film School|url=https://www.sherwoodoaksfilmschool.com/our-history/,%20https://www.sherwoodoaksfilmschool.com/our-history/|access-date=2020-08-18|language=en-US}}{{Dead link|date=March 2023 |bot=InternetArchiveBot |fix-attempted=yes }}&lt;/ref&gt; Screenwriting teacher [[Syd Field]] conducted his first workshop there.&lt;ref&gt;{{Cite news|url=https://sydfield.com/products/screenplay-the-foundations-of-screenwriting/|title=Screenplay - Syd Field|work=Syd Field|access-date=2017-09-10|language=en-US}}&lt;/ref&gt;{{Failed verification|date=August 2020}} Through the years hundreds of industry professionals have lectured at the school, including Paul Newman, Robert Redford, George C. Scott, Lily Tomlin, Mel Brooks, Shelley Winters, Robert Altman, Jack Lemmon, and Lucille Ball.&lt;ref name=&quot;:1&quot; /&gt; Notable alumni include Sylvester Stallone, James Cameron, Paul Haggis, and Nancy Meyers.&lt;ref name=&quot;:2&quot; /&gt;

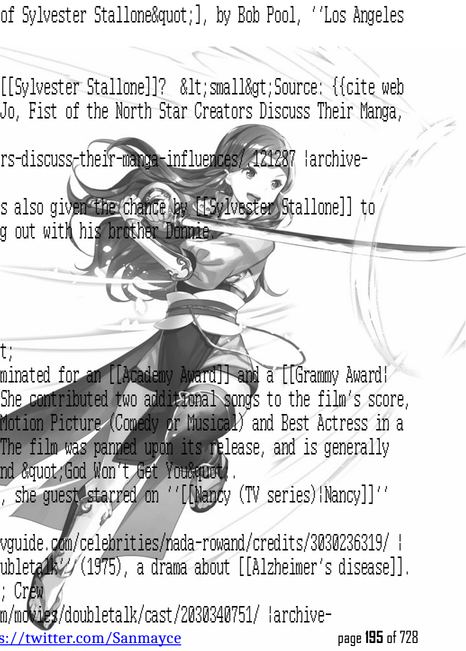
3030 :Typically South Philly Residents are portrayed By New York actors, like Sylvestor Stallone, who played Rocky, from Philly. There are also highly distinct class differences in the eastern cities.

3031 *'''Born''' : [[Sage Stallone]], American actor and filmmaker, son of [[Sylvester Stallone]]; in [[Los Angeles]] (d. 2012 of coronary artery disease)&lt;ref&gt;[https://www.latimes.com/local/obituaries/la-me-0714-sage-stallone-20120714-story.html &quot;Sage Stallone dies at 36; son of Sylvester Stallone&quot;], by Bob Pool, ''Los Angeles Times'', July 14, 2012&lt;/ref&gt;

3032   <title>Sylvester Stallone (song)</title>

3033 ** '''ALT1''':... that the muscular character designs in the manga '''''[[Phantom Blood]]''''' were inspired by [[Arnold Schwarzenegger]] and [[Sylvester Stallone]]? &lt;small&gt;Source: {{cite web |url=http://www.animenewsnetwork.com/interest/2017-09-13/jojo-fist-of-the-north-star-creators-discuss-their-manga-influences/.121287 |title=JoJo, Fist of the North Star Creators Discuss Their Manga, Influences |last=Whalen |first=Amanda |date=2017-09-13 |website=[[Anime News Network]] |publisher=2017-09-13 |access-date=2017-09-30 |archive-url=https://web.archive.org/web/20170914123847/http://www.animenewsnetwork.com/interest/2017-09-13/jojo-fist-of-the-north-star-creators-discuss-their-manga-influences/.121287 |archive-date=2017-09-14 |url-status=live}}&lt;/small&gt;

3034 After engaging in a mixed wrestling/boxing match with [[Andre the Giant]], a down on his luck Wepner meets a woman called Linda in a bar. He is also given the chance by [[Sylvester Stallone]] to audition for a role in [[Rocky II]], though following a confrontation with the producer, his part gets cut. He subsequently gets into a falling out with his brother Donnie.



3035 *[[Anthony Mangano]] as [[Sylvester Stallone]]

3036 [[Category:Cultural depictions of Sylvester Stallone]]

3037 |[[Sylvester Stallone]]

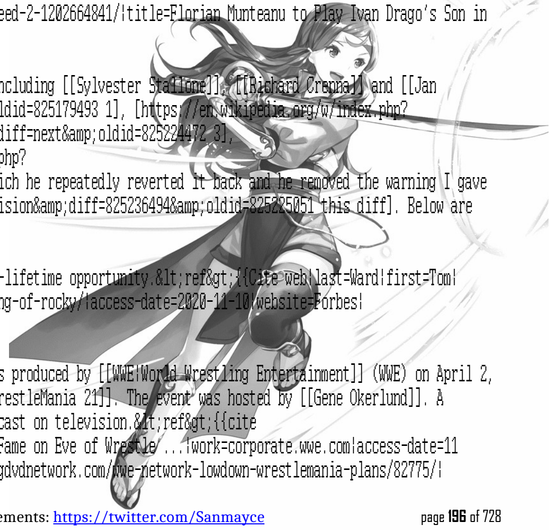
3038 * [[Sylvester Stallone]] (2008)

3039 Hence, Por Ruamrudee earned the nickname &quot;Rambo&quot; after the [[Sylvester Stallone]] [[John Rambo|movie character]].&lt;ref name=era/&gt;

3040 In 1980 Parton made her theatrical film debut as Doralee Rhodes in ''[[9 to 5 (film)|9 to 5]]''. She wrote the film's theme song, which was nominated for an [[Academy Award]] and a [[Grammy Award|Grammy]]. In 1982 Parton appeared in ''[[The Best Little Whorehouse in Texas (film)|The Best Little Whorehouse in Texas]]'' as Mona Stangley. She contributed two additional songs to the film's score, &quot;Sneakin' Around&quot; and &quot;I Will Always Love You&quot;. The film and Parton herself were nominated for [[Golden Globes]] for Best Motion Picture (Comedy or Musical) and Best Actress in a Motion Picture (Comedy or Musical), respectively. In 1984 Parton starred in ''[[Rhinestone (film)|Rhinestone]]'' with [[Sylvester Stallone]]. The film was panned upon its release, and is generally regarded as a commercial and critical flop. However, the soundtrack yielded two top 10 hits for Parton, &quot;Tennessee Homesick Blues&quot; and &quot;God Won't Get You&quot;.

3041 In 1970 and 1972, she performed in [[United Service Organizations|USO]] tours in Vietnam.&lt;ref name=&quot;soapbook&quot; /&gt; On television, she guest starred on ''[[Nancy (TV series)|Nancy]]'' and ''[[Bewitched]]''.&lt;ref name=&quot;soapbook&quot; /&gt;&lt;ref name=&quot;credits&quot;&gt;{{Cite web |title=Credits: Nada Rowand |url=https://www.tvguide.com/celebrities/nada-rowand/credits/3030236319/ |archive-url=https://web.archive.org/web/20241001184355/https://www.tvguide.com/celebrities/nada-rowand/credits/3030236319/ | archive-date=October 1, 2024 |access-date=October 1, 2024 |website=[[TV Guide (magazine)|TV Guide]]}}&lt;/ref&gt; Rowand played Mother in ''Doubletalk'' (1975), a drama about [[Alzheimer's disease]]. The production was nominated for an [[Academy Award for Best Live Action Short Film]].&lt;ref&gt;{{Cite web |title=Doubletalk - Full Cast &amp; Crew |url=https://www.tvguide.com/movies/doubletalk/cast/2030340751/ |archive-url=https://web.archive.org/web/20241001185618/https://www.tvguide.com/movies/doubletalk/cast/2030340751/ |archive-

```
     date=October 1, 2024 |access-date=October 1, 2024 |website=[[TV Guide (magazine)|TV Guide]]}}&lt;/ref&gt; She also appeared in the film ''Super Seal'' (1975), co-starring with [[Sterling Holloway]].&lt;ref&gt;{{Cite web |title=Super Seal |url=https://www.tcm.com/tcmdb/title/498487/super-seal#credits |archive-url=https://web.archive.org/web/20241001190441/https://www.tcm.com/tcmdb/title/498487/super-seal#credits |archive-date=October 1, 2024 |access-date=October 1, 2024 |website=[[Turner Classic Movies]]}}&lt;/ref&gt; In 1978, she co-starred with [[Sylvester Stallone]] in the film ''[[F.I.S.T. (film)|F.I.S.T.]]''&lt;ref&gt;{{Cite web |title=F.I.S.T. |url=https://www.tcm.com/tcmdb/title/22946/fist#credits |archive-url=https://web.archive.org/web/20220610080027/https://www.tcm.com/tcmdb/title/22946/fist#credits |archive-date=June 10, 2022 |access-date=October 1, 2024 |website=[[Turner Classic Movies]]}}&lt;/ref&gt; Rowand returned to Broadway in 1979, as a &quot;stand by&quot; for the roles of Elizabeth and The Duchess of York in a revival of ''[[Richard III (play)|Richard III]]'', starring [[Al Pacino]].&lt;ref name=&quot;nada&quot;&gt;{{Cite web |title=Nada Rowand |url=https://m.playbill.com/person/nada-rowand-vault-0000049757 |archive-url=https://web.archive.org/web/20241001191601/https://m.playbill.com/person/nada-rowand-vault-0000049757 |archive-date=October 1, 2024 |access-date=October 1, 2024 |website=[[Playbill]]}}&lt;/ref&gt;
3042 The tower, located on {{convert|4|acre|abbr=on}},&lt;ref name=Hundreds/&gt; was [[Topping out|topped off]] on July 2, 2008, during a ceremony that included actor [[Sylvester Stallone]] and Nevada governor [[Jim Gibbons (American politician)|Jim Gibbons]] as guests.&lt;ref name=Jul08&gt;{{cite news|last=Clarke|first=Norm|title=Playboy models in state spotlight|url=https://www.reviewjournal.com/news/playboy-models-in-state-spotlight/|access-date=October 24, 2017|work=Las Vegas Review-Journal|date=July 3, 2008|quote=Sylvester Stallone and Gibbons, among dignitaries at Wednesday's topping-off of David Siegel's PH Towers by Westgate. A 50-foot American flag was unfurled atop the tower as Lee Greenwood sang his &quot;God Bless the USA&quot; and the national anthem.}}&lt;/ref&gt; PH Towers' opening had initially been set for August 21, 2009.&lt;ref name=Oct2012&gt;{{cite news|last=O'Reiley|first=Tim|title=Opening statements aired over construction of PH Towers Westgate|url=https://www.reviewjournal.com/crime/courts/opening-statements-aired-over-construction-of-ph-towers-westgate/|access-date=October 26, 2017|work=Las Vegas Review-Journal|date=October 8, 2012}}&lt;/ref&gt; In July 2009, Planet Hollywood began hiring 800 workers for PH Towers, which was to have a [[soft opening]] that October, followed by the official grand opening a month later.&lt;ref name=Finnegan&gt;{{cite news|last=Finnegan|first=Amanda|title=Planet Hollywood hiring 800 workers for new towers|url=https://lasvegassun.com/news/2009/jul/22/planet-hollywood-seeking-800-employees/|access-date=October 24, 2017|work=Las Vegas Sun|date=July 22, 2009}}&lt;/ref&gt;&lt;ref name=Hundreds&gt;{{cite news|last=Knightly|first=Arnold M.|title=Hundreds of job seekers arrive for Planet Hollywood job fair|url=https://www.reviewjournal.com/business/hundreds-of-job-seekers-arrive-for-planet-hollywood-job-fair/|access-date=October 24, 2017|work=Las Vegas Review-Journal|date=July 31, 2009}}&lt;/ref&gt; By late September 2009, the tower's opening was scheduled for December. At the time, two additional towers were planned to be opened by November 2013.&lt;ref name=Knightly&gt;{{cite news|last=Knightly|first=Arnold M.|title=Planet Hollywood, Westgate bringing 1,201 rooms to Strip with PH Towers|url=https://www.reviewjournal.com/business/planet-hollywood-westgate-bringing-1201-rooms-to-strip-with-ph-towers/|access-date=October 24, 2017|work=Las Vegas Review-Journal|date=September 29, 2009}}&lt;/ref&gt; The first tower was completed at an ultimate cost of $660 million. Construction never began on additional towers.&lt;ref name=Troubled/&gt;
3043 # {{Icon|B}} [[Sylvester Stallone]]
3044 * In an ''[[Access (U.S. TV series)|Access Hollywood]]'' interview, [[Sylvester Stallone]], who had also launched his own bottled water line called Sly Water, was asked to take a blind taste test of two types of water: his and that of a competitor. When asked which one he preferred after tasting both, Stallone correctly identified his own water, and added he would not wash his socks using the other water. When told that the other water was Trump Ice, Stallone laughed and apologized to Donald Trump, before quipping that ''[[Rocky]] [[Rocky (film series)|VII]]'' would be a showdown between his and Trump's brands of water.&lt;ref&gt;{{cite AV media |people=Sylvester Stallone|date=May 4, 2007|title=Stallone Water?|medium=Television show|language=en |url=https://www.youtube.com/watch?v=EZwo0xlnFhM|access-date=December 6, 2015}}&lt;/ref&gt;
3045 * 02 Sep 2018 &amp;ndash; [[:Category:Soundtracks to films directed by Sylvester Stallone]] CfDed by [[User:Koavf|Koavf]] was closed; [[Wikipedia:Categories for discussion/Log/2018_September_2#Category:Soundtracks to films by director|discussion]]
3046 | Although most Americans likely never heard of her, Sridevi had an amazing career in [[Cinema of India|Indian cinema]]. The so-called &quot;first female superstar&quot; of [[Bollywood]] starred in [[Sridevi filmography|300 films]] in her 51-year career, capping it off with her critically acclaimed performance in [[Mom (film)|''Mom'']] that won the [[National Film Award for Best Actress]]. Sadly, the award had to be given posthumously, as Sridevi drowned in a hotel room in [[Dubai]] on February 24th. Rumors originally circulated that the death was an [[hoax#types|internet hoax]], like the hoax that had circulated the a few days before about [[Sylvester Stallone]] (who barely missed making this list), but her brother-in-law [[Sanjay Kapoor]] soon confirmed the death to the media. She was given a full [[state funeral#India|state funeral]], rare for non-politicians, and her [[funeral procession]] attracted thousands of mourners.
3047 |[[Sylvester Stallone]]&lt;br /&gt;''[[Creed (film)|Creed]]''
3048 Raspberry Blood and Oscar247 both edited Sylvester Stalone, though years apart.
3049 | Guests          = Day 1 of 12 Days, [[Sylvester Stallone]]
3050 | Guests          = [[Sylvester Stallone]], [[Gabrielle Reece]], [[Abbi Jacobson]] &amp; [[Ilana Glazer]]
3051 | ''[[Guardians of the Galaxy Vol. 2]]'' || [[Marvel Studios]] || [[James Gunn]] (director/screenplay); [[Chris Pratt]], [[Zoe Saldaña]], [[Dave Bautista]], [[Vin Diesel]], [[Bradley Cooper]], [[Michael Rooker]], [[Karen Gillan]], [[Pom Klementieff]], [[Sylvester Stallone]], [[Kurt Russell]] || &lt;ref&gt;{{cite web|title=Guardians of the Galaxy Vol. 2|url=https://in.bookmyshow.com/movies/guardians-of-the-galaxy-vol-2/ET00048079|website=BookMyShow|access-date=February 5, 2017}}&lt;/ref&gt;
3052 I tried to post sexual assault allegations against [[Sylvester Stallone]] on his page. I used BBC News as my source (http://www.bbc.co.uk/news/entertainment-arts-42023885), however another user removed it for being unreliable because the article mentions that the police report was published by the Daily Mail.
3053 *'''Delete'''. No evidence of notability in either English or Persian. Refs #1, #3, #4, #5, are the personal website of the subject. Refs #2,#6, #7 (websavar.ir) link to a page that does not appear to contain his name. Ref #8 is an article about famous Western actors such as Bruce Willis and Sylvester Stallone.  A blatant deceptive use of a ref. Ref #9 has no usable content. Ref #10 again does not contain his name. [[User:Spinningspark|&lt;b style=&quot;background:#FAFAD2;color:#C08000&quot;&gt;Spinning&lt;/b&gt;]][[User talk:Spinningspark|&lt;b style=&quot;color:#4840A0&quot;&gt;Spark&lt;/b&gt;]] 02:43, 11 March 2018 (UTC)
3054 '''Jane Wooster Scott''' is an American painter,&lt;ref name=&quot;Sky Auction&quot;&gt;{{cite web|title=A Rambunctious Razzle Dazzle LIMITED EDITION Lithograph by Jane Wooster Scott! Numbered and Hand Signed with Certificate of Authenticity!|url=http://www.skyauction.com/anyoffer/121471|publisher=Sky Auction|accessdate=4 February 2018}}&lt;/ref&gt; who was named by [[Guinness Book of World Records]] as the most reproduced artist in America.&lt;ref name=&quot;Art Deals&quot;&gt;{{cite web|title=Jane Wooster Scott Artist Biography and Art Gallery Collection|url=https://artdeals.co/collections/jane-wooster-scott-artist-biography-and-art-gallery-collection|publisher=Art Deals|accessdate=4 February 2018}}&lt;/ref&gt;  She started painting at the age of 35 and has never had a lesson.&lt;ref name=&quot;Game Wright&quot;&gt;{{cite web|title=Meet the Artist: Jane Wooster Scott|date=17 April 2012 |url=https://gamewright.wordpress.com/2012/04/17/meet-the-artist-jane-wooster-scott/|publisher=Game Wright|accessdate=4 February 2018}}&lt;/ref&gt;  Her works hang on permanent display at the American embassies in Portugal and Australia, and have hung at the US [[White House]].&lt;ref name=&quot;King Features&quot;&gt;{{cite web|title=Acclaimed Artist, Jane Wooster Scott, Represented For Licensing By King Features|date=28 June 2001 |url=http://kingfeatures.com/2001/06/acclaimed-artist-jane-wooster-scott-represented-for-licensing-by-king-features/|publisher=King Features|accessdate=4 February 2018}}&lt;/ref&gt; She has sold works to [[Carol Burnett]], [[Charlton Heston]], [[Paul Newman]], [[Arnold Schwarzenegger]], [[Nancy Sinatra]],&lt;ref name=&quot;People&quot;&gt;{{cite web|last1=Hoover|first1=Eleanor|title=Jane Wooster Scott Is the Improbable Grandma Moses of Hollywood Hills|url=http://people.com/archive/jane-wooster-scott-is-the-improbable-grandma-moses-of-hollywood-hills-vol-8-no-24/|publisher=People|accessdate=4 February 2018}}&lt;/ref&gt; [[Sylvester Stallone]] and many others.&lt;ref name=&quot;MG Puzzles&quot;&gt;{{cite web|title=Jane Wooster Scott|url=https://www.mgcpuzzles.com/mgcpuzzles/artgallery/Jane_Wooster_Scott/index.htm|publisher=MG Puzzles|accessdate=4 February 2018}}&lt;/ref&gt;
3055 Caple's next film was the 2018 sports action drama film ''[[Creed II]]'', a sequel to 2015 film ''[[Creed (film)|Creed]]'' and the eighth installment in the [[Rocky (film series)|''Rocky'' film series]].&lt;ref name=DeadlineCreed2&gt;{{cite news|title='Creed 2' Taps Steven Caple Jr. To Replace Sylvester Stallone As Director|url=https://deadline.com/2017/12/creed-2-steven-caple-jr-sylvester-stallone-directing-1202224671/|accessdate=February 4, 2018|publisher=Deadline.com|date=December 11, 2017}}&lt;/ref&gt;  Stallone said of appointing Caple as director: “I believe it's important for the director to also be a part of this generation like I was in mine, to make the story as relatable as possible. We are extremely lucky to have the talented young filmmaker Steven Caple Jr. step up and accept the role of director. I am confident that he and Michael B. Jordan will hit it out of the park!”&lt;ref name=VarietyCreed2&gt;{{cite news|title=Steven Caple Jr. to Direct 'Creed 2' Starring Michael B. Jordan and Sylvester Stallone|url=https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/#article-comments|accessdate=February 4, 2018|publisher=Variety.com|date=December 11, 2017}}&lt;/ref&gt; Production of ''Creed II'' began in spring 2018. The film was released on November 21, 2018 and received positive reviews.&lt;ref name=&quot;VarietyCreed2release&quot;&gt;{{cite news|url=https://variety.com/2018/film/news/florian-munteanu-ivan-dragos-son-in-creed-2-1202664841/|title=Florian Munteanu to Play Ivan Drago's Son in 'Creed 2'|date=January 15, 2018|accessdate=February 4, 2018|publisher=Variety.com}}&lt;/ref&gt;
3056 | [[Sylvester Stallone]]
3057 [[User talk:Joefromrandb|Joefromrandb]] has been commiting disruptive editing by removing the U.S. marks from various infoboxes, including [[Sylvester Stallone]], [[Richard Crenna]] and [[Jan Maxwell]], among others, as seen in those diffs [https://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=prev&amp;oldid=825179493 1], [https://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=next&amp;oldid=825203481 2], [https://en.wikipedia.org/w/index.php?title=Sylvester_Stallone&amp;diff=next&amp;oldid=825224472 3],[https://en.wikipedia.org/w/index.php?title=Richard_Crenna&amp;diff=prev&amp;oldid=825179709 4],[https://en.wikipedia.org/w/index.php?title=Jan_Maxwell&amp;type=revision&amp;diff=825221953&amp;oldid=825218464 5]. I reverted two of his edit on Sylvester Stallone which he repeatedly reverted it back and he removed the warning I gave him about it as seen on [https://en.wikipedia.org/w/index.php?title=User_talk%3AJoefromrandb&amp;action=historysubmit&amp;type=revision&amp;diff=825236494&amp;oldid=825225051 this diff]. Below are pagelinks of pages involve in Joefromrandb's disruptive editing, among others.
3058 {{pagelinks|Sylvester Stallone}}
3059 |[[Academy Award]] winner for Best Picture about [[Rocky Balboa|a Philadelphia pug]] ([[Sylvester Stallone]]) who gets a once-in-a-lifetime opportunity.&lt;ref&gt;{{Cite web|last=Ward|first=Tom|title=The Amazing Story Of The Making Of 'Rocky'|url=https://www.forbes.com/sites/tomward/2017/08/29/the-amazing-story-of-the-making-of-rocky/|access-date=2020-11-10|website=Forbes|language=en}}&lt;/ref&gt;
3060 |Washed-up boxers ([[Robert De Niro]], [[Sylvester Stallone]]) square off one last time.
3061 '''WWE Hall of Fame (2005)''' was the event which featured introduction of the 6th class to the [[WWE Hall of Fame]]. The event was produced by [[WWE|World Wrestling Entertainment]] (WWE) on April 2, 2005, from the [[Universal Amphitheatre]] in [[Los Angeles|Los Angeles, California]]. The event took place the same weekend as [[WrestleMania 21]]. The event was hosted by [[Gene Okerlund]]. A condensed version of the ceremony aired that evening on [[Paramount Network|Spike TV]], making this ceremony the first to be broadcast on television.&lt;ref&gt;{{cite web|url=http://corporate.wwe.com/news/company-news/2005/03-21-2005|title=Sylvestor Stallone to Induct Hulk Hogan into WWE Hall of Fame on Eve of Wrestle ...|work=corporate.wwe.com|access-date=11 April 2018}}&lt;/ref&gt; In March 2015 the ceremony was added to the [[WWE Network]].&lt;ref&gt;{{cite web|url=http://www.wrestlingdvdnetwork.com/wwe-network-lowdown-wrestlemania-plans/82775/|title=WWE Network Lowdown: WrestleMania Plans, Free Gifts, 'Every' Hall of Fame?|access-date=11 April 2018}}&lt;/ref&gt;
```



3062 |[[Sylvester Stallone]]

3063 Having signed with Scotti Brothers Records in 1985,&lt;ref&gt;{{cite web|author=Biography by Tom Demalon |url=https://www.allmusic.com/artist/robert-tepper-mn0000291687/biography |title=Robert Tepper &amp;#124; Biography &amp; History |publisher=AllMusic |access-date=2018-04-11}}&lt;/ref&gt; Tepper's debut album was preceded by the title track, which [[Sylvester Stallone]] had heard and chose to feature on the soundtrack of his film ''[[Rocky IV]]''.&lt;ref&gt;{{cite web|author=Rob Theakston |url=https://www.allmusic.com/album/rocky-iv-original-motion-picture-soundtrack-mw0000191595 |title=Rocky IV [Original Motion Picture Soundtrack&amp;#93; - Various Artists &amp;#124; Songs, Reviews, Credits |publisher=AllMusic |access-date=2018-04-11}}&lt;/ref&gt;&lt;ref name=&quot;archive1&quot;&gt;{{cite web |url=http://www.aor.nu:80/index.php?option=com_content&amp;task=view&amp;id=52&amp;Itemid=90 |archive-url=https://web.archive.org/web/20071128103935/http://www.aor.nu/index.php?option=com_content&amp;task=view&amp;id=52&amp;Itemid=90 |url-status=dead |archive-date=2007-11-28 |title=aor.nu - Robert Tepper |date=2007-11-28 |access-date=2018-04-11 }}&lt;/ref&gt; &quot;Angel of the City&quot; was also featured on the soundtrack of Stallone's 1986 film ''[[Cobra (1986 film)|Cobra]]''.&lt;ref&gt;{{cite web|url=https://www.allmusic.com/album/cobra-mw0000189418 |title=Cobra - Original Soundtrack &amp;#124; Songs, Reviews, Credits |publisher=AllMusic |access-date=2018-04-11}}&lt;/ref&gt;

3064 {{Wikipedia:Wikipedia Signpost/Templates/Filler image-v2|image=File:Sylvester Stallone 2012.jpg|size=250px|caption=The [[List of premature obituaries#Sylvester Stallone|reports of my death]] have been greatly exaggerated.}}

3065 | [[Sylvester Stallone]]

3066 Her first movie role was as the housemaid of a Mafia don, played by [[Sylvester Stallone]], in the movie ''[[Oscar (1991 film)|Oscar]]'' (1991).&lt;ref name=&quot;:0&quot; /&gt;&lt;ref&gt;{{Cite news |url=https://www.latimes.com/archives/la-xpm-1991-04-20-ca-58-story.html |title=Joycelyn O'Brien's First Movie Is With Sly Stallone, Thank You |last=KING |first=SUSAN |date=1991-04-20 |work=Los Angeles Times |access-date=2018-03-27 |language=en-US |issn=0458-3035}}&lt;/ref&gt; She also appeared in ''[[The Mambo Kings]]'' (1992) as Gina.

3067 *[[Sylvester Stallone]] (1997)

3068 Known as &quot;The Legendary South Beach Gatekeeper&quot;.[1], Cedric Adegnika (born August 3, 1973) moved from France to the US in 1995. He worked at the most prestigious venues in the country such as Bar None (owned by Sylvester Stallone), The Living Room, Man Ray, Mynt, Set, LIV and Story [2]. Throughout his career he denied access to celebrities such as Dennis Rodman [3], Wilmer Valderrama, Jamie Foxx (with whom he later became friend)&amp; Robert De Niro establishing a reputation for himself. With a resume that includes events during the Cannes Film Festival &amp; Paris Fashion Week, and two years at the A-list restaurant-lounge Man Ray in Chelsea New York City (owned by Johnny Depp, Sean Penn, John Malkovich, and Harvey Weinstein) [4]. It was here, eighteen years after moving to Miami and living the South Beach lifestyle, that he realized his influence. In 2013 he left his velvet rope days far behind and opened THE FLAT, a cocktail lounge which quickly became a hotspot [5], and despite his success Adegnika sold it to move to Panama City Panama where he opened LESSEP'S bistro cafe [6]

3069 | style=&quot;text-align: center;&quot;| &lt;ref&gt;{{Cite news|url=https://www.yahoo.com/entertainment/viola-davis-sylvester-stallone-tv-084428046.html|title=Viola Davis and Sylvester Stallone's TV Pilot We All Wish We Could Have Seen|access-date=May 13, 2018|language=en-US}}&lt;/ref&gt;

3070 *[[Paul Bloch]], 78, American publicist ([[Eddie Murphy]], [[Bruce Willis]], [[Sylvester Stallone]]).&lt;ref&gt;[https://people.com/movies/paul-bloch-dies-at-78/ Hollywood Publicist Paul Bloch, Whose Clients Included Eddie Murphy and Bruce Willis, Dies at 78]&lt;/ref&gt;

3071 The artistic capability and spirit of innovation of the mask artists come to the fore when unusual characters have to be made. In recent past, masks of Marvel comic characters such as [[Green Goblin (Ultimate Marvel character)|Green Goblin]], Sylvester Stallone, or [[Wolverine (character)|Wolverine]] have been created by artists, and these are particularly popular among the non-chhau buyers.&lt;ref name=&quot;:5&quot; /&gt;

3072 Noël Carroll references Mulvey's pivotal paper on psychoanalysis and visual pleasure in his writing, and plays devil's advocate to her claim that women are the only subjects of gaze. Carroll acknowledges and agrees with Mulvey's assessment that women in film are strategically placed for the male gaze despite the role of their actual character. Carroll states, &quot;Women in Hollywood film are staged and blocked for male erotic contemplation and pleasure.&quot;&lt;ref name=&quot;:10&quot;&gt;{{cite journal |jstor=431572|title=The Image of Women in Film: A Defense of a Paradigm|journal=The Journal of Aesthetics and Art Criticism|volume=48|issue=4|pages=349-360|last1=Carroll|first1=Noël|year=1990|doi=10.2307/431572}}&lt;/ref&gt; However, Carroll adds that men in films are also strategically placed for the purpose of pleasure. He cites such examples as [[Sylvester Stallone]] and [[Arnold Schwarzenegger]], big bodybuilding actors &quot;whose scenes are blocked and staged precisely to afford spectacles of bulging pectorals and other parts.&quot;&lt;ref name=&quot;:10&quot; /&gt; Similarly to actresses, male actors are also heralded for their facial attractiveness and are sometimes lauded exclusively for being attractive. As an example, Carroll offers [[Leslie Howard]], a male actor who appeared in ''[[Of Human Bondage (1934 film)|Of Human Bondage]]'' and ''[[Gone with the Wind (film)|Gone With the Wind]],'' who was highly successful in the industry despite being &quot;staggeringly ineffectual.&quot;&lt;ref name=&quot;:10&quot; /&gt; According to Carroll, being subject to the erotic gaze of the audience is not an exclusively female burden; rather, both sexes fall prey to Hollywood camera angles that best show off their bodies.

3073 '''''Champion''''' is a 2018 South Korean [[Sports film|sports]] [[comedy film]] directed by Kim Yong-wan. The film stars [[Ma Dong-seok]], [[Kwon Yul (actor)|Kwon Yul]], and [[Han Ye-ri]]. It is related to arm wrestling competition. The movie is rated 9.4/10 in Asian TV Website within the first month of release.&lt;ref&gt;{{cite news|url=https://www.hancinema.net/kwon-yul-and-han-ye-ri-to-star-in-champion-109729.html|title=Kwon Yul and Han Ye-ri to star in &quot;Champion&quot;|date=4 June 2017|work=Joy News|publisher=[[HanCinema]]|accessdate=May 22, 2018}}&lt;/ref&gt;&lt;ref&gt;{{cite news|last1=Yoon|first1=Min-sik|title='Mavely' shadows hero Sylvester Stallone in arm wrestling film 'Champion'|url=http://www.koreaherald.com/view.php?ud=20180402000812|accessdate=May 22, 2018|work=[[The Korea Herald]]|date=April 2, 2018}}&lt;/ref&gt;&lt;ref&gt;Champion (2018) Movie, Asiantv Website, available at http://myasiantv.se/movie/champion/&lt;/ref&gt;

3074 Collectors and critics have noted his bold and rich palettes throughout his career.&lt;ref name=&quot;Philadelphia Originals&quot;&gt;{{cite book |last1=Glantz |first1=Joseph |title=Philadelphia Originals |date=2009 |publisher=Schiffer Publishing |location=Atglen, PA |isbn=978-0-7643-3338-5 }}&lt;/ref&gt; His list of commission patrons and collectors includes names such as [[Sylvester Stallone]], [[Julius Erving]], the [[Philadelphia Museum of Art]], the [[Philadelphia Eagles]]&lt;ref name=&quot;Legendary Locals of Center City Philadelphia&quot;&gt;{{cite book |last1=Nickels |first1=Thom |title=Legendary Locals of Center City Philadelphia |date=2014 |publisher=Arcadia Publishing |location=Charleston, SC |isbn=978-1-4671-0141-7 }}&lt;/ref&gt; and [[Ed Rendell]].

3075 His portfolio's long list of recognizable figures includes names such as [[Liz Taylor]], [[Marilyn Monroe]], [[Frida Kahlo]], [[Geronimo]], characters from television's ''[[The Sopranos]]'',&lt;ref name=&quot;Legendary Locals of Center City Philadelphia&quot; /&gt; [[Frank Sinatra]], [[George W. Bush]],&lt;ref name=&quot;Patriotic Art Becomes 'Melting Pot'&quot; /&gt; Rocky Balboa,&lt;ref name=&quot;Philadelphia Originals&quot; /&gt; the [[Dalai Lama]], [[Nelson Mandela]], [[Sophia Loren]], [[Jackie Kennedy]], [[Robin Williams]], [[Madonna (entertainer)|Madonna]], [[Barbra Streisand]], Sylvester Stallone, [[Elvis Presley]], [[Lady Gaga]], [[Chazz Palminteri]], [[Cole Hamels]], [[Bruce Springsteen]], and [[Albert Einstein]].&lt;ref name=&quot;The Delaware Velley Italian-American Herold&quot;&gt;{{cite news |last1=Zippi |first1=Barbara Ann |title=Italian Culture, Inside the colorful world of the man who painted the pope |volume=2 |issue=9 |publisher=The Delaware Velley Italian-American Herold |date=September 2015}}&lt;/ref&gt; In 2007 his Marilyn Monroe series was featured in the ''Marylin in Art'' book published by Cantor Publishing Group.&lt;ref name=&quot;Philadelphia Originals&quot; /&gt; In January 2017, Milou presented [[Joe Biden]] his portrait of the former US Vice President upon his return home to Delaware after the [[inauguration]].&lt;ref name=&quot;Carson Wentz meets Philly artist who painted his portrait&quot;&gt;{{cite news |last1=Koller |first1=Brock |title=Carson Wentz meets Philly artist who painted his portrait |url=http://6abc.com/sports/wentz-meets-philly-artist-who-painted-his-portrait/2090454/ |accessdate=2 June 2018 |agency=6ABC News |publisher=6ABC News |date=June 14, 2017}}&lt;/ref&gt;

3076 {{Short description|Action-thriller film series starring Sylvester Stallone}}

3077 The '''''Escape Plan''''' film series,&lt;ref name=&quot;Series_SF&quot;&gt;{{cite web|url=https://www.slashfilm.com/1508321/correct-order-to-watch-escape-plan-movies/|work=Slash Film|title=The Correct Order to Watch the Escape Plan Movies|author=Schaefer, Sandy|date=February 2, 2024|accessdate=August 24, 2024}}&lt;/ref&gt; consists of American [[prison film|prison]] [[Action film|action]]-[[Thriller film|thrillers]] based on characters created by Miles Chapman and [[Jason Keller (playwright)|Arnell Jesko]]. The series is centered on Ray Breslin, a [[security]] analyst and [[Escapology|escape artist]] who helps design [[supermax prison]]s by posing as an inmate to find their flaws; later films center on his security and [[hostage]] rescue firm. The series stars [[Sylvester Stallone]], with a supporting cast including [[Arnold Schwarzenegger]], [[Dave Bautista]] and [[50 Cent|Curtis &quot;50 Cent&quot; Jackson]] as allies and associates of Breslin. All entries in the series since the second film have primarily been released [[direct-to-video]], though they have also been released in some theaters. The series has received a mixed-to-negative critical response.

3078 | colspan=&quot;3&quot; | [[Sylvester Stallone]]

3079 | &lt;ref name=&quot;Escape Plan: The Extractors_BoxOfficeMojo&quot;&gt;{{Cite web|url=https://www.boxofficemojo.com/release/rl4277699585/weekend/|title=Escape Plan: The Extractors|website=Box Office Mojo}}&lt;/ref&gt;&lt;ref name=&quot;Escape Plan: The Extractors_TheNumbers&quot;&gt;{{Cite web|url=https://www.the-numbers.com/movie/Escape-Plan-The-Extractors|title=Escape Plan: The Extractors (2019) - Financial Information|website=The Numbers}}&lt;/ref&gt;&lt;ref name=&quot;Stallone is Proud&quot;&gt;{{cite web|url=https://www.cinemablend.com/news/2476477/sylvester-stallone-is-so-proud-of-escape-plan-3-success-since-it-was-made-in-just-17-days|title=Sylvester Stallone Is 'So Proud' Of Escape Plan 3 Success, Since It Was Made In Just 17 Days|date=2019-07-14|publisher=CinemaBlend|access-date=2019-11-17}}&lt;/ref&gt;

3080 | occupation = Kickboxing promoter, television producer,&lt;ref&gt;{{Cite news|url=http://www.mediafax.ro/sport/eduard-irimia-am-avut-nas-fin-florian-munteanu-a-fost-reperat-de-sylvester-stallone-17233449|title=Eduard Irimia: Am avut nas fin. Florian Munteanu a fost reperat de Sylvester Stallone|work=[[Mediafax]]|date=28 May 2018|language=Romanian}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=https://www1.agerpres.ro/sport/2013/9/18/Arte-martiale-Eduard-Irimia-va-produce-cel-mai-mare-festival-oriental-din-Europa18-09-2013-15-55-15|title=Arte marţiale: Eduard Irimia va produce cel mai mare festival oriental din Europa|work=[[Agerpres]]|date=18 September 2013|language=Romanian}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=https://www.cancan.ro/eduard-irimia-dat-lovitura-la-hollywood-cu-fabrica-de-vedete-19767764|title=Eduard Irimia a dat lovitura la Hollywood cu &quot;fabrica de vedete&quot;!|work=cancan.ro|date=3 June 2018|language=Romanian}}&lt;/ref&gt;&lt;ref&gt;{{Cite news|url=https://www.click.ro/sport/alte-sporturi/romanul-care-se-trage-de-sireturi-cu-sylvester-stallone-face-furori-la-bucuresti|title=Românul care se trage de şireturi cu Sylvester Stallone face furori la Bucureştia|work=click.ro|date=29 May 2018|language=Romanian}}&lt;/ref&gt;   businessman

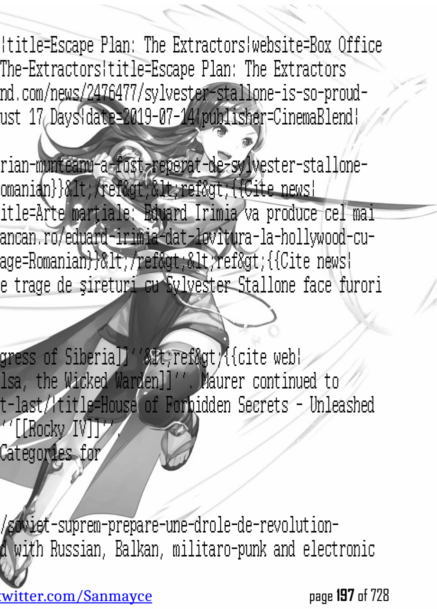
3081 |[[Sylvester Stallone]]

3082 |url = https://books.google.com/books?id=DoUHPAAACAAJ|pages = 142|publisher = Telos Publishing|isbn = 9781845830304}}&lt;/ref&gt; ''[[Ilsa, the Tigress of Siberia]]''&lt;ref&gt;{{cite web|url=http://www.cinepix.ca/cinepix-horror-rama/|title=Cinepix at Horror -Rama}}&lt;/ref&gt; and director [[Jess Franco]]'s unofficial entry, ''[[Ilsa, the Wicked Warden]]''. Maurer continued to appear in film and television, both alongside Thorne&lt;ref&gt;{{cite web|url=https://houseofforbiddensecrets.wordpress.com/2013/10/14/unleashed-at-last/|title=House of Forbidden Secrets - Unleashed at Last}}&lt;/ref&gt; and on his own, including experimental anthology film ''[[Aria (1987 film)|Aria]]'' and [[Sylvester Stallone]] boxing film, ''[[Rocky IV]]''.

3083 * 02 Sep 2018 &amp;ndash; [[:Category:Soundtracks to films directed by Sylvester Stallone]] CfDed by [[User:Koavf|Koavf]] was closed; [[Wikipedia:Categories for discussion/Log/2018_September_2#Category:Soundtracks to films by director|discussion]]

3084 | alias   = R.wan / Sylvester Staline

3085 He also took part in the group Soviet Suprem where he was known as the eccentric &quot;Sylvester Staline&quot;.&lt;ref&gt;[https://www.humanite.fr/soviet-suprem-prepare-une-drole-de-revolution-musicale-578830 ''L'Humanité'': Soviet Suprem prépare une drole de révolution musicale]&lt;/ref&gt; The group takes Soviet-themed songs often mixed with Russian, Balkan, militaro-punk and electronic music.



3086 The collective members are Erwan Séguillon (better known as R.wan from band Java) - known as the character &quot;Sylvester Staline&quot;&lt;ref&gt;[http://www.sovietsuprem.com/sylvester-staline/ Soviet Supre website: Sylvester Staline page] {{in lang¦fr}}&lt;/ref&gt; in the formation and Tom Feterman (from the band La Caravane Passe) known in Soviet Supreme as the character &quot;John Lénine&quot;. The band also includes the characters &quot;DJ Croute Chef&quot; and &quot;Yougo Chavez&quot;.

3087 ¦ ''[[The Suicide Squad (film)¦The Suicide Squad]]'' ¦¦ [[Warner Bros. Pictures]] / [[DC Films]] / [[Atlas Entertainment]] / [[HBO Max]] ¦¦ [[James Gunn]] (director/screenplay); [[Idris Elba]], [[Margot Robbie]], [[John Cena]], [[Joel Kinnaman]], [[Sylvester Stallone]], [[Viola Davis]], [[David Dastmalchian]], [[Daniela Melchior]], [[Michael Rooker]], [[Jai Courtney]], [[Peter Capaldi]], [[Alice Braga]], [[Pete Davidson]], [[Nathan Fillion]], [[Sean Gunn]], [[Flula Borg]], Mayling Ng ¦¦style=&quot;text-align:center;&quot;¦&lt;ref&gt;{{cite web¦last=Melendez¦first=Marcos¦date=July 9, 2021¦title=New 'The Suicide Squad' Poster and Early Preview Screening Date Revealed by James Gunn¦url=https://collider.com/the-suicide-squad-poster-early-screening-date-james-gunn/¦website=[[Collider (website)¦Collider]]¦url-status=live¦archive-url=https://web.archive.org/web/20210710010030/https://collider.com/the-suicide-squad-poster-early-screening-date-james-gunn/¦archive-date=July 10, 2021¦access-date=July 11, 2021}}&lt;/ref&gt;

3088 In 2006 Petruccio created a series of fine art originals in connection with [[Metro-Goldwyn-Mayer¦MGM]] and [[Sylvester Stallone]] to commemorate the 30th Anniversary of [[Rocky]]. In 2008 The Hard Rock Theme Park announced Petruccio as their official artist tasked to create artistic imagery throughout the park. In 2009 Muhammad Ali Enterprises named Petruccio their approved official artist.&lt;ref&gt;{{Cite news¦url=http://www.nydailynews.com/sports/baseball/mets/new-york-mets-fan-joe-petruccio-draws-inspiration-sketches-daily-news-cartoonist-bill-gallo-article-1.181916#¦title=New York Mets fan Joe Petruccio draws inspiration for sketches from Daily News cartoonist Bill Gallo - NY Daily News¦last=LEMIRE¦first=JONATHAN¦work=nydailynews.com¦access-date=2018-08-19}}&lt;/ref&gt;

3089 *Lieutenant Marion &quot;Cobra&quot; Cobretti, in the movie ''[[Cobra (1986 film)¦Cobra]]'', played by [[Sylvester Stallone]]

3090 *[[Sylvester Stallone¦Michael Sylvester Gardenzio Stallone]], birth name of American actor and film director Sylvester Stallone

3091 Along with being announced as the director of ''The Heavy'', a Paramount and Bad Robot production, Avery had been recruited by [[20th Century Fox]] to write and direct a [[Flash Gordon (film)#Reboot¦remake of Flash Gordon]] alongside [[Matthew Vaughn]] who was to be producing.&lt;ref&gt;{{cite web¦author=Dave McNary ¦url=https://www.variety.com/2018/film/news/overlord-director-flash-gordon-movie-1203015289/amp/ ¦title='Overlord' Director Julius Avery Boards 'Flash Gordon' Movie - Variety ¦publisher=Variety.com ¦date=2018-10-30 ¦accessdate=2018-11-20}}&lt;/ref&gt;&lt;ref&gt;{{cite news¦url=https://www.hollywoodreporter.com/amp/heat-vision/flash-gordon-movie-be-directed-by-julius-avery-1156437¦title='Flash Gordon' Movie to Be Directed by Julius Avery¦work=Hollywood Reporter ¦ date=30 October 2018 ¦ first=Aaron ¦ last=Couch ¦ accessdate=18 November 2019 }}&lt;/ref&gt; However, following [[Acquisition of 21st Century Fox by Disney¦Disney's acquisition of 21st Century Fox's assets]], the film was cancelled. He instead directed MGM's ''[[Samaritan (film)¦Samaritan]]'', a dark superhero film starring [[Sylvester Stallone]], released in August 26, 2022 it went onto to be the number one watched movies on [[Prime Video]] that year.&lt;ref&gt;{{cite web¦last=D'Alessandro¦first=Anthony¦title=Sylvester Stallone MGM Thriller 'Samaritan' Heads To June 4, 2021¦url=https://deadline.com/2019/11/sylvester-stallone-mgm-thriller-samaritan-opening-late-fall-2020-1202743588/¦date=27 November 2019¦website=[[Deadline Hollywood]]¦access-date=8 December 2019}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦last=Vitale ¦first=Micaela Pérez ¦date=2023-06-24 ¦title=The Most Streamed Prime Video Movies and TV Shows in 2022 ¦url=https://movieweb.com/prime-video-movies-tv-series-most-watched-2022-amazon/ ¦access-date=2024-02-06 ¦website=MovieWeb ¦language=en}}&lt;/ref&gt;

3092 ¦[[Sylvester Stallone]]

3093 ¦The latest in the ''[[Rocky (film series)¦Rocky]]'' film series, starring [[Michael B. Jordan]] (pictured) as [[Adonis Creed]] and [[Sylvester Stallone]] as his trainer [[Rocky Balboa]], ''Creed II'' sees Creed go up against Viktor Drago, son of the [[Ivan Drago¦Russian boxer]] who killed [[Apollo Creed]], Adonis's father, in a [[Rocky IV¦prior instalment]]. The film has received positive reviews, and good box office.

3094 ¦rowspan=2¦ [[Sylvester Stallone]] (death hoax)

3095 ¦ ''[[Creed II]]'' ¦¦ [[Metro-Goldwyn-Mayer]] / [[Warner Bros. Pictures]] ¦¦ [[Steven Caple Jr.]] (director); [[Juel Taylor]], [[Sylvester Stallone]] (screenplay); [[Michael B. Jordan]], Sylvester Stallone, [[Tessa Thompson]], [[Wood Harris]], [[Phylicia Rashad]], [[Florian Munteanu]], [[Dolph Lundgren]] ¦¦&lt;ref&gt;{{cite news ¦first=Justin ¦last=Kroll ¦title=Steven Caple Jr. to Direct 'Creed 2' Starring Michael B. Jordan and Sylvester Stallone (EXCLUSIVE) ¦url=https://variety.com/2017/film/news/creed-2-director-steven-caple-jr-1202626542/ ¦work=[[Variety (magazine)¦Variety]] ¦access-date=December 11, 2017}}&lt;/ref&gt;

3096 Later in his career, Barker focused much more on studies of media audiences, drawing upon his background in [[cultural studies]]. His co-authored study of audiences of the [[Sylvester Stallone]] film, ''[[Judge Dredd (film)¦Judge Dredd]]'', was influential in the field of audience studies. In response to the censorship campaign against [[David Cronenberg]]'s ''[[Crash (1996 film)¦Crash]]'', again by the [[Daily Mail]], Barker, along with Jane Arthurs and Ramiswami Harindranath, oversaw and authored a study on the film's reception in the press and conducted an audience study. This was published as ''The Crash Controversy: Censorship Campaigns and Film Reception'' by Wallflower Press in 2001. Over the course of these and other audience studies, Barker developed a methodology for studying audiences which to date remains very influential in the field.

3097 ¦ Although most Americans likely never heard of her, Sridevi had an amazing career in [[Cinema of India¦Indian cinema]]. The so-called &quot;first female superstar&quot; of [[Bollywood]] starred in [[Sridevi filmography¦300 films]] in her 51-year career, capping it off with her critically-acclaimed performance in [[Mom (film)¦''Mom'']] that won the [[National Film Award for Best Actress]]. Sadly, the award had to be given posthumously, as Sridevi drowned in a hotel room in [[Dubai]] on February 24th. Rumors originally circulated that the death was an [[hoax#types¦internet hoax]], like the hoax that had circulated the a few days before about [[Sylvester Stallone]] (who barely missed making this list), but her brother-in-law [[Sanjay Kapoor]] soon confirmed the death to the media. She was given a full [[state funeral#India¦state funeral]], rare for non-politicians, and her [[funeral procession]] attracted thousands of mourners.

3098 ¦Guests=[[Sylvester Stallone]], [[Cedric the Entertainer]]

3099 ¦ starring   = [[Sylvester Stallone]] &lt;br /&gt;[[Ryan Guzman]] &lt;br /&gt;[[Meadow Williams]] &lt;br /&gt;[[Christopher McDonald]] &lt;br /&gt;[[Colin Egglesfield]] &lt;br /&gt;Lydia Hull &lt;br /&gt;Tyler Jon Olson &lt;br /&gt;Sergio Rizzuto &lt;br /&gt;Swen Temmel &lt;br /&gt;[[Matthew Modine]]

3100 '''''Backtrace''''' is a 2018 American [[action film¦action]] [[crime film]] directed by Brian A. Miller, and starring [[Sylvester Stallone]], [[Matthew Modine]] and [[Ryan Guzman]]. The film was released as [[video on demand]] and limited theatrical release by [[Lionsgate Premiere]] on December 14, 2018 in the United States.&lt;ref&gt;{{Cite web¦url=https://www.comingsoon.net/movies/trailers/1007693-1007693¦title = New Trailer for Sylvester Stallone's Backtrace Brings the Pain to VOD¦date = 13 November 2018}}&lt;/ref&gt;&lt;ref&gt;{{cite web¦url=https://rogersmovienation.com/2018/12/07/movie-review-stallone-is-back-on-the-beat-with-backtrace/¦title=Movie Review: Stallone is back on the beat with &quot;Backtrace&quot;¦date=December 7, 2018}}&lt;/ref&gt;

3101 After suffering a brain injury from a [[bank heist]] gone wrong, MacDonald (Matthew Modine) develops [[amnesia]] and is put into a prison psychiatric ward. Following his seventh year in evaluation, he is coerced by an inmate and a ward doctor (Ryan Guzman and Meadow Williams) to break out of prison and injected with a serum that forces him to relive the life he's forgotten. MacDonald must now elude a local detective (Sylvester Stallone), a toughened FBI agent (Christopher McDonald) and the drug's dangerous side effects in order to recover the stolen money all while confronting his past.

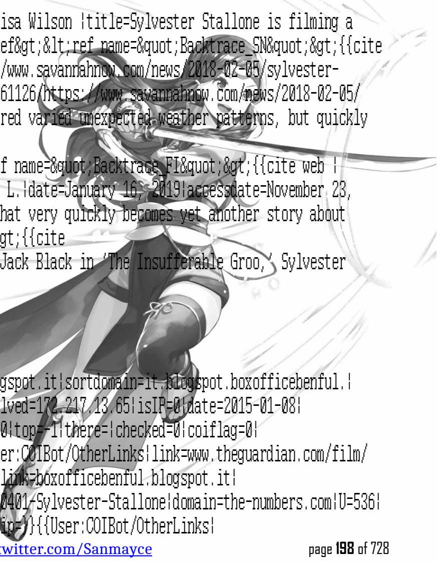
3102 * [[Sylvester Stallone]] as Detective Sykes: An experienced police detective, assigned to work with Agent Franks on the case involving Donovan MacDonald.

3103 ''Backtrace'' was officially announced in November 2018, co-starring [[Sylvester Stallone]], [[Matthew Modine]] and [[Ryan Guzman]]. Brian A. Miller served as director, with a script written by Mike Maples, the story was detailed as an action crime-thriller involving Stallone's Det. Sykes pursuit of an escaped convicted psychotic bank robber.&lt;ref name=&quot;Backtrace_MW&quot;&gt;{{cite web¦url=https://movieweb.com/backtrace-trailer-sylvester-stallone/¦work=MovieWeb¦title=Backtrace Trailer Has Stallone Hunting a Psychotic Bank Robber¦author=Orange, B. Alan¦date=November 13, 2018¦accessdate=November 23, 2021}}&lt;/ref&gt;

3104 Miller described the project as a [[whodunit]]-styled crime mystery. The filmmaker described his experiences with the cast with high praise, noting his life-long fandom of Sylvester Stallone. Miller stated that Stallone's experiences in filmmaking enhanced the film, classifying Stallone as an [[auteur]]. The filmmaker additionally that though the final act is a plot-twist, clues and hints are scattered throughout the movie, intended for repeated viewings from the audience. The script underwent several rewrites, prior to production, with major plot changes; while McDonald, Guzman, and Egglesfield joined production immediately after receiving word that Stallone had signed on to co-star in the film.&lt;ref name=&quot;Backtrace_Storyocity&quot;&gt;{{cite web¦url=https://www.storyocity.co/blog/2018/12/17/story-behind-the-story-backtrace¦work=Storyocity¦title=Story Behind the Story: Backtrace¦author=Storyocity¦date=December 20, 2018¦accessdate=November 23, 2021}}&lt;/ref&gt;

3105 The film was shot in [[Savannah, Georgia]] and lasted 12 days.&lt;ref name=&quot;Backtrace_IP&quot;&gt;{{cite web ¦date=February 6, 2018 ¦author=Lisa Wilson ¦title=Sylvester Stallone is filming a movie in Savannah. Here's what we know about it ¦url=https://www.islandpacket.com/entertainment/movies-news-reviews/article198604349.html }}&lt;/ref&gt;&lt;ref name=&quot;Backtrace_SN&quot;&gt;{{cite web ¦date=2018-02-05 ¦last=Rayburn ¦first=Josh ¦title=Sylvester Stallone, Matthew Modine, Colin Egglesfield movie filming in Savannah ¦url=https://www.savannahnow.com/news/2018-02-05/sylvester-stallone-matthew-modine-colin-egglesfield-movie-filming-savannah ¦website=Savannah Morning News ¦archive-url=https://web.archive.org/web/20180211161126/https://www.savannahnow.com/news/2018-02-05/sylvester-stallone-matthew-modine-colin-egglesfield-movie-filming-savannah ¦archive-date=2018-02-11 ¦url-status=dead}}&lt;/ref&gt; Production endured varied unexpected weather patterns, but quickly completed principal photography.&lt;ref name=&quot;Backtrace_Storyocity&quot; /&gt;

3106 Kevin L. Lee of ''Film Inquiry'' called the film a &quot;cinematic...headache.&quot; The critic criticized the writing, and the film's tone.&lt;ref name=&quot;Backtrace_FI&quot;&gt;{{cite web ¦url=https://www.filminquiry.com/backtrace-2018-review/¦title=BACKTRACE: The Cinematic Equivalent Of A Headache¦work=Film Inquiry¦author=Lee, Kevin L.¦date=January 16, 2019¦accessdate=November 23, 2021}}&lt;/ref&gt; Similarly, Noel Murray of the ''[[Los Angeles Times]]'' critiqued the plot as unoriginal and familiar, stating: &quot;A movie that very quickly becomes yet another story about people with guns chasing other people with guns, through featureless forests and abandoned buildings.&quot;&lt;ref name=&quot;Backtrace_LAT&quot;&gt;{{cite web¦url=https://www.latimes.com/entertainment/movies/la-et-mn-capsule-genre-string-20181212-story.html ¦work=The Los Angeles Times¦title=Reviews: Jack Black in 'The Insufferable Groo,' Sylvester Stallone (barely) in 'Backtrace' and other movies¦author=Murray, Noel¦date=Dec 13, 2018 ¦accessdate=November 23, 2021}}&lt;/ref&gt;



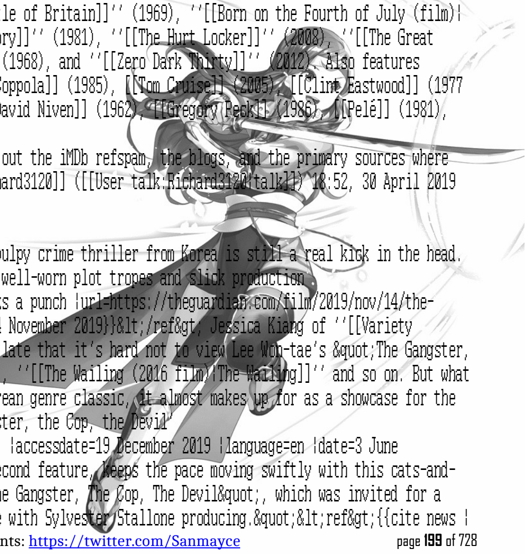
3107 ** [[Sylvester Stallone]] - ''[[Rambo: Last Blood]]'' as [[John Rambo¦John J. Rambo]]

3108 ** [[Sylvester Stallone]] and his impotent rage - ''[[Rambo: Last Blood]]''

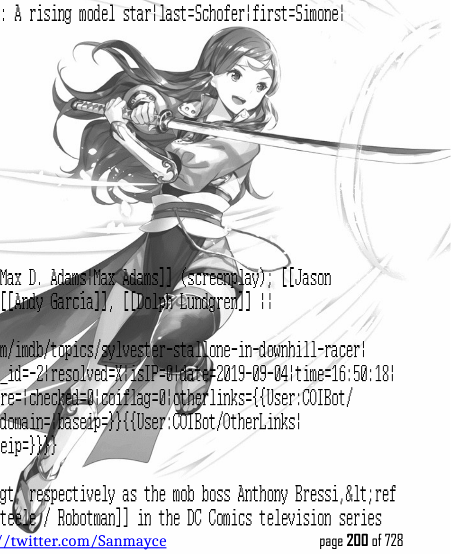
3109 ** ''[[Rambo: Last Blood]]'' - Matthew Cirulnick and [[Sylvester Stallone]]; based on the [[John Rambo¦character]] created by [[David Morrell]]

3110 {{User:COIBot/EditSummary¦id=43178964¦lang=it¦wikidomain=w¦namespace=¦pagename=Sylvester Stallone¦username=Martin Mystère¦link=boxofficebenful.blogspot.it¦sortdomain=it.blogspot.boxofficebenful.¦domain=boxofficebenful.blogspot.it¦origdiff=http://it.wikipedia.org/w/index.php?diff=70107480&amp;oldid=70071259&amp;rcid=86827065¦edit_id=-2¦resolved=172.217.13.65¦isIP=0¦date=2015-01-08¦time=14:22:20¦wiki=it.wikipedia.org¦revid=70107480¦oldid=70071259¦usercount=536¦whitelisted=0¦blacklisteduser=0¦whitereason=¦blackreason=¦deleted=0¦top=-1¦there=¦checked=0¦coiflag=0¦base=boxofficebenful.blogspot.com/¦basedomain=boxofficebenful.blogspot.com¦baseip=172.217.13.65¦remark=&lt;!-- redirectsite --&gt;¦otherlinks={{User:COIBot/OtherLinks¦link=www.theguardian.com/film/2013/nov/27/fifth-estate-2103-biggest-movie-turkey¦domain=theguardian.com¦U=536¦L=¦UL=X¦WUL=X¦base=¦basedomain=¦baseip=}}{{User:COIBot/OtherLinks¦link=boxofficebenful.blogspot.it¦domain=boxofficebenful.blogspot.it¦U=536¦L=58¦UL=X¦WUL=X¦base=¦basedomain=¦baseip=}}{{User:COIBot/OtherLinks¦link=www.the-numbers.com/person/135950401-Sylvester-Stallone¦domain=the-numbers.com¦U=536¦L=-1¦UL=X¦WUL=X¦base=¦basedomain=¦baseip=}}{{User:COIBot/OtherLinks¦link=www.imdb.com¦domain=imdb.com¦U=536¦L=-1¦UL=X¦WUL=X¦base=¦basedomain=¦baseip=}}{{User:COIBot/OtherLinks¦

```
link=boxofficemojo.com/people/chart/?view=Actor&amp;amp;id=sylvesterstallone.htm|domain=boxofficemojo.com|U=536|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.youtube.com/watch?v=ty3_i7IWmg8|domain=youtube.com|U=536|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=uk.askmen.com/entertainment/top5/9.html|domain=uk.askmen.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=guide.supereva.it/registi_cinema/interventi/2010/07/sylvester-stallone-happy-birthday-6-giugno-2010|domain=guide.supereva.it|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=moviemavericks.com/2010/08/stallone-becomes-1st-movie-star-to-headline-1-openings-in-last-5-decades-news/|domain=moviemavericks.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.takesontech.com/?tag=sylvester-stallone|domain=takesontech.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.cinezapping.com/2010/07/24/il-panel-di-the-expendables-al-comic-con-e-tre-clip-inedite/|domain=cinezapping.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.hollywoodreporter.com/news/acapulco-film-fest-honor-sylvester-672289|domain=hollywoodreporter.com|U=536|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.trud.bg/Article.asp?ArticleId=2427066|domain=trud.bg|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.ilsole24ore.com/art/cultura/2012-11-09/sylvester-stallone-lupa-capitolina-183739.shtml?uuid=AbsphY1G|domain=ilsole24ore.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=eagleandbadgefoundation.com/news/2012_eagle_and_badge_gala_-_saturday_october_6_2012/|domain=eagleandbadgefoundation.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.express.co.uk/posts/view/349803/Sylvester-Stallone-to-be-honoured-for-police-support|domain=express.co.uk|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.hollywoodreporter.com/news/cinemacon-2012-sylvester-stallone-feted-career-achievement-315032|domain=hollywoodreporter.com|U=536|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.cinemacon.com/schedule/2012-events/|domain=cinemacon.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=sylvesterstallone.com/sly-receives-lifetime-achievement-award-2/|domain=sylvesterstallone.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.variety.com/article/VR1118029154|domain=variety.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=popwatch.ew.com/2010/12/07/sylvester-stallone-boxing-hall-of-fame/|domain=popwatch.ew.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.hollywoodnews.com/2010/10/04/sylvester-stallone-morgan-freeman-lori-mccreary-and-director-tom-hooper-to-be-honored-at-the-hollywood-awards-gala/|domain=hollywoodnews.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.youtube.com/watch?v=sYKNgPdbTAw|domain=youtube.com|U=536|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=uk.ign.com/action-hero/sylvester-stallone.html|domain=uk.ign.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=corp.ign.com/articles/110/1107676p1.html|domain=corp.ign.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.cinemaitaliano.info/news/03686/a-sylvester-stallone-il-jaeger-lecoultre.html|domain=cinemaitaliano.info|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.imdb.com/event/ev0001506/2008|domain=imdb.com|U=536|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.eastsideboxing.com/news.php?p=6661&amp;amp;more=1|domain=eastsideboxing.com|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.bwaa.org/2005_stallone.htm|domain=bwaa.org|U=536|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.imdb.com/name/nm0000230/awards|domain=imdb.com|U=536|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}}}
3111    <redirect title="Sylvester Stallone" />
3112      <comment>[[User:Cewbot/log/20201008/configuration|Fixing broken anchor]]: #Early life→most alike anchor [[Sylvester Stallone#Early life and education]]</comment>
3113      <text bytes="130" sha1="9ftojg4uewkudlinxruy0img00x3dcc" xml:space="preserve">#REDIRECT [[Sylvester Stallone#Early life and education]]
3114 | [[1976 in film|1976]] || ''[[Rocky]]'' || $225,000,000 ($77,100,000){{ref|Rentals|R}} || Boxing || $960,000 || &lt;ref name=&quot;RoNu&quot;/&gt;&lt;ref&gt;{{cite web|url=https://www.youtube.com/watch?v=Rru5cJYKH80|title=Sylvester Stallone on the Tonight Show Starring Johnny Carlson|website=YouTube|date=2 March 2020 |access-date=August 2, 2020|archive-date=August 19, 2020|archive-url=https://web.archive.org/web/20200819121801/https://www.youtube.com/watch?v=Rru5cJYKH80&amp;gl=US&amp;hl=en|url-status=live}}&lt;/ref&gt;
3115 Ali's next opponents included [[Chuck Wepner]], [[Ron Lyle]], and [[Joe Bugner]]. Wepner, a journeyman known as &quot;The Bayonne Bleeder&quot;, stunned Ali with a &quot;knockdown&quot; in the ninth round; Ali would later say he tripped on Wepner's foot, or that Wepner had actually stepped on Ali's foot and he lost his balance. Wepner also admitted that is what happened in a later interview. {{citation needed|date=January 2024}} It was a bout that would inspire [[Sylvester Stallone]] to create the acclaimed film, ''[[Rocky]]''.&lt;ref&gt;{{Cite web|url=https://www.forbes.com/2006/08/10/rocky-stallone-settlement-cx_rs_0810autofacescan04|title=Stallone Settles With The 'Real' Rocky|first=R. M.|last=Schneiderman|website=Forbes|access-date=April 18, 2019|archive-date=April 18, 2019|archive-url=https://web.archive.org/web/20190418224955/https://www.forbes.com/2006/08/10/rocky-stallone-settlement-cx_rs_0810autofacescan04|url-status=live}}&lt;/ref&gt;
3116 The fight took place on October 2, 1980, in Las Vegas Valley, with Holmes easily dominating Ali, who was weakened from thyroid medication he had taken to lose weight. Giachetti called the fight &quot;awful ... the worst sports event I ever had to cover.&quot; Actor [[Sylvester Stallone]] was at ringside and said that it was like watching an autopsy on a man who is still alive.&lt;ref name=&quot;Hauser 2004&quot;/&gt; In the eleventh round, Angelo Dundee told the referee to stop the fight, making it the only time that Ali ever lost by [[technical knockout|stoppage]]. After the fight, Holmes went back to his dressing room and cried. The Holmes fight is said to have contributed to Ali's Parkinson's syndrome.&lt;ref&gt;{{cite news | url=https://www.nytimes.com/2009/10/27/arts/television/27muhammad.html | work=The New York Times | title=Boxing King Casts His Shadow, Even at Time of Defeat | date=October 26, 2009 | access-date=March 5, 2012 | first1=Mike | last1=Hale | archive-date=March 10, 2012 | archive-url=https://web.archive.org/web/20120310053609/http://www.nytimes.com/2009/10/27/arts/television/27muhammad.html? | url-status=live }}&lt;/ref&gt; Despite pleas to definitively retire, Ali fought one last time on December 11, 1981, in Nassau, [[Bahamas]], against [[Trevor Berbick]], losing a ten-round decision.&lt;ref&gt;{{cite news |title=Ali to try again? |url=https://news.google.com/newspapers?nid=1870&amp;dat=19810816&amp;id=9_EwAAAAIBAJ&amp;pg=2603,113296&amp;hl=en |newspaper=The Daytona Beach Sunday News-Journal |agency=Associated Press |date=August 16, 1981 |access-date=June 4, 2016 |archive-date=August 28, 2021 |archive-url=https://web.archive.org/web/20210828083317/https://news.google.com/newspapers?nid=1870&amp;dat=19810816&amp;id=9_EwAAAAIBAJ&amp;pg=2603%2C113296&amp;hl=en |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite news |title=It's all over for Ali after loss |url=https://news.google.com/newspapers?nid=2199&amp;dat=19811212&amp;id=XKUyAAAAIBAJ&amp;pg=3700,2405394&amp;hl=en |newspaper=Lawrence Journal-World |agency=Associated Press |date=December 12, 1981 |access-date=June 4, 2016 |archive-date=September 29, 2021 |archive-url=https://web.archive.org/web/20210929102632/https://news.google.com/newspapers?nid=2199&amp;dat=19811212&amp;id=XKUyAAAAIBAJ&amp;pg=3700,2405394&amp;hl=en |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite magazine |first=William |last=Nack |author-link=William Nack |title=Not with a bang but a whisper |url=https://www.si.com/vault/1981/12/21/826243/not-with-a-bang-but-a-whisper-after-losing-to-trevor-berbick-a-subdued-muhammad-ali-softly-admitted-that-his-illustrious-career-had-come-to-an-end |magazine=Sports Illustrated |date=December 21, 1981 |access-date=June 4, 2016 |archive-date=June 9, 2016 |archive-url=https://web.archive.org/web/20160609212952/http://www.si.com/vault/1981/12/21/826243/not-with-a-bang-but-a-whisper-after-losing-to-trevor-berbick-a-subdued-muhammad-ali-softly-admitted-that-his-illustrious-career-had-come-to-an-end |url-status=live }}&lt;/ref&gt;
3117    <title>Category:Cultural depictions of Sylvester Stallone</title>
3118      <text bytes="385" sha1="nmf4swvsr5bk8ucgbi3czllf0jqdoys" xml:space="preserve">A list of cultural depictions of [[Sylvester Stallone]].
3119 [[Category:Sylvester Stallone|Cultural depictions]]
3120 | {{Won}}&lt;ref&gt;Hollywood Awards (4 October 2010). &quot;[http://hollywoodawards.com/news/2010/hooper-sly-innovator-release.html Sylvester Stallone, Morgan Freeman, Lori Mccreary and Director Tom Hooper to be Honored at the Hollywood Awards Gala]&quot;. Press release. Retrieved 6 October 2010 ([https://web.archive.org/web/20101230041800/http://hollywoodawards.com/news/2010/hooper-sly-innovator-release.html archived] by WebCite on 3 January 2011).&lt;/ref&gt;
3121 ** [[Sylvester Stallone]], American actor and film director (voice of Corporal Weaver in ''[[Antz]]'', Lieutenant Victor von Ion in ''[[Ratchet &amp; Clank (film)|Ratchet &amp; Clank]]'', Bulletman in ''[[Animal Crackers (2017 film)|Animal Crackers]]'', [[Paul Revere]] in the ''[[Liberty's Kids]]'' episode &quot;Midnight Ride&quot;).
3122 * '''August 9''': [[Alain Dorval]], French voice actor (French dub voice of Pete in the [[Mickey Mouse|''Mickey Mouse'' franchise]], Weaver in ''[[Antz]]'', Tiger in the [[An American Tail (franchise)|''American Tail'' franchise]], Victor in ''[[Ratchet &amp; Clank (film)|Ratchet &amp; Clank]]'', Lex Luthor in ''[[Superman: The Animated Series]]'', Sylvester Stallone in ''[[Solar Opposites]]'', voice of Goliath the Rhino in ''[[The Jungle Bunch (film)|The Jungle Bunch]]''), (d. [[2024]]).&lt;ref&gt;{{Cite web |date=2024-02-13 |title=Mort du comédien Alain Dorval, la voix française de Sylvester Stallone |url=https://www.francetvinfo.fr/culture/cinema/mort-du-comedien-alain-dorval-la-voix-francaise-de-sylvester-stallone_6363346.html |access-date=2024-02-13 |website=Franceinfo |language=fr-FR}}&lt;/ref&gt;&lt;ref&gt;[https://www.libramemoria.com/deces-celebres/2024/02/13/le-doubleur-francais-alain-dorval-est-decede Le doubleur français Alain Dorval est décédé] {{in lang|fr}}&lt;/ref&gt;&lt;ref&gt;[https://www.leparisien.fr/culture-loisirs/cinema/alain-dorval-voix-francaise-de-sylvester-stallone-et-pere-de-la-ministre-aurore-berge-est-mort-a-77-ans-13-02-2024-LBRHLKLMV5EZXE4PCFFCEHLJYY.php Alain Dorval, voix française de Sylvester Stallone et père de la ministre Aurore Bergé, est mort à 77 ans] {{in lang|fr}}&lt;/ref&gt;
3123 ''[[1917 (2019 film)|1917]]'' (2019), ''[[Apocalypse Now]]'' (1979), ''[[American Sniper]]'' (2014), ''[[Battle of Britain (film)|Battle of Britain]]'' (1969), ''[[Born on the Fourth of July (film)|Born on the Fourth of July]]'' (1989), ''[[The Dirty Dozen]]'' (1967), ''[[Dunkirk (2017 film)|Dunkirk]]'' (2017), ''[[Escape to Victory]]'' (1981), ''[[The Hurt Locker]]'' (2008), ''[[The Great Escape (film)|The Great Escape]]'' (1963), ''[[The Guns of Navarone (film)|The Guns of Navarone]]'' (1961), ''[[Where Eagles Dare]]'' (1968), and ''[[Zero Dark Thirty]]'' (2012). Also features excerpts of interviews with [[Kathryn Bigelow]] (2013), [[Richard Burton]] (1977), [[Michael Caine]] (1969 and 1981), [[Francis Ford Coppola]] (1985), [[Tom Cruise]] (2005), [[Clint Eastwood]] (1977 and 2015), [[Stewart Granger]] (1981), [[Ron Kovic]] (1980), [[Lee Marvin]] (1980), [[Sam Mendes]] (2019), [[Bobby Moore]] (1981), [[David Niven]] (1962), [[Gregory Peck]] (1986), [[Pelé]] (1981), [[Sylvester Stallone]] (1981 and 1985), [[John Sturges]] (1989), and [[Harry Styles]] with [[Fionn Whitehead]] (2017).}}
3124 *'''Delete''': fails [[WP:GNG]] and [[WP:ACTOR]]. Simply no independent reliable sources other the reports about his court case. Take out the iMDb refspam, the blogs, and the primary sources where Mr. With discusses Sylvester Stallone, his TEDex talks and his car, and there is nothing of substantial value. [[User:Richard3120|Richard3120]] ([[User talk:Richard3120|talk]]) 18:52, 30 April 2019 (UTC)
3125 [[Sylvester Stallone]]&lt;br&gt;
3126 Leslie Felperin of ''[[The Guardian]]'' stated &quot;Although its final act shreds credulity, and the structure is a bit wonky, this pulpy crime thriller from Korea is still a real kick in the head. Like so many of the genre that hail from Seoul and its suburbs, this one punches well above its weight with an inventive reworking of well-worn plot tropes and slick production values.&quot;&lt;ref&gt;{{cite web |last1=Felperin |first1=Leslie |title=The Gangster, the Cop, the Devil review - pulpy thriller packs a punch |url=https://theguardian.com/film/2019/nov/14/the-gangster-the-cop-the-devil-review-lee-won-tae-korea-crime-thriller |website = [[The Guardian]] | accessdate=19 December 2019 | date=14 November 2019}}&lt;/ref&gt; Jessica Kiang of ''[[Variety (magazine)|Variety]]'' commented &quot;Korea has dominated the midnight-movie/genre slots at international festivals so thoroughly of late that it's hard not to view Lee Won-tae's &quot;The Gangster, the Cop, the Devil&quot; in terms of its shortcomings in comparison to the likes of ''[[Train to Busan]]'', ''[[The Age of Shadows]]'', ''[[The Wailing (2016 film)|The Wailing]]'' and so on. But what this fun, slick but slightly forgettable hardboiled actioner lacks in terms of the energy, originality and inventiveness of a true Korean genre classic, it almost makes up for as a showcase for the burly charisma of star Don Lee, aka Ma Dong-Seok.&quot;&lt;ref&gt;{{cite web |last=Kiang |first=Jessica |title=Film Review: 'The Gangster, the Cop, the Devil' |url=https://variety.com/2019/film/festivals/the-gangster-the-cop-the-devil-review-1203229234/ |website=[[Variety (magazine)|Variety]] |accessdate=19 December 2019 |language=en |date=3 June 2019}}&lt;/ref&gt; Cary Darling of the ''[[Houston Chronicle]]'' added &quot;Director/writer Lee Won-tae, for whom this is only his second feature, keeps the pace moving swiftly with this cats-and-mouse game, showing off an energetic sensibility that heralds a new voice on the South Korean film scene. It's no wonder that &quot;The Gangster, The Cop, The Devil&quot;, which was invited for a midnight-screening slot at the recently concluded Cannes Film Festival, has been picked up by Hollywood for an English-language remake with Sylvester Stallone producing.&quot;&lt;ref&gt;{{cite news |
```



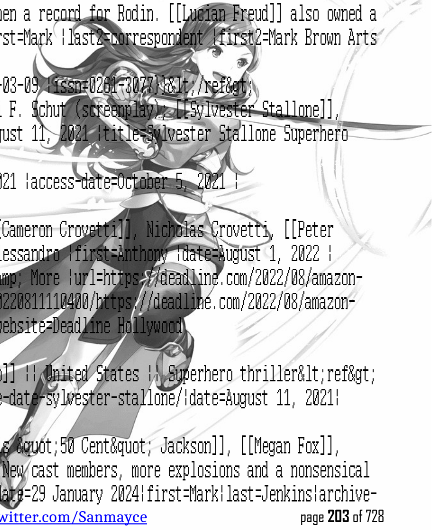
```
     last1=Darling |first1=Cary |title='Gangster' should make Don Lee a star |url=https://www.houstonchronicle.com/entertainment/movies/article/Gangster-should-make-Don-Lee-a-star-13934748.php | website=[[Houston Chronicle]] |accessdate=19 December 2019 |date=4 June 2019}}&lt;/ref&gt;
3127 On 5 May 2019, it was announced that [[Sylvester Stallone]] and his [[Balboa Productions]] partner Braden Aftergood will produce the US remake of ''The Gangster, the Cop, the Devil'', with [[Ma Dong-seok]] reprising his role and producing the film under BA Entertainment.&lt;ref&gt;{{cite web|first=Lee|last=Hyo-won|title=Sylvester Stallone to Remake South Korean Thriller 'The Gangster, The Cop, The Devil'|url=https://www.hollywoodreporter.com/news/balboa-productions-remake-south-korean-thriller-gangster-cop-devil-1207585|date=5 May 2019|website=[[The Hollywood Reporter]]|access-date=9 May 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last1=BECHERVAISE |first1=JASON |title='The Gangster, The Cop, The Devil': Cannes Review |url=https://www.screendaily.com/reviews/the-gangster-the-cop-the-devil-cannes-review/5139312.article |website=[[Screen Daily]] |access-date=19 December 2019 |date=15 May 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last1=Howard |first1=Kirsten |title=The Gangster, The Cop, The Devil review |url=https://www.filmstories.co.uk/reviews/the-gangster-the-cop-the-devil-review/ |website=Film Stories |access-date=19 December 2019|date=13 November 2019}}&lt;/ref&gt;
3128 Voice actors Ray Chase and Max Mittelman were cast to play George and Steven, respectively. Also cast were television actor Matthew Bohrer (John Williams), Lex Lang (Harrison Ford), Julia McIlvaine (Marcia Lucas), and Kenny Holmes (Sylvester Stallone).
3129 |[[Sylvester Stallone]]
3130 *Miz Cracker - [[Sylvester Stallone]]
3131 '''Lee Won-tae''' is a South Korean film director, screenwriter and producer. Formerly working as a producer on programs for [[MBC TV (South Korean TV channel)|MBC TV]], Lee debuted with the period prison drama ''[[Man of Will]]'' in 2017. Prior to that, he also served as producer on the 2011 horror romcom ''[[Spellbound (2011 film)|Spellbound]]'' and penned the original story for the 2015 period film ''[[The Magician (2015 film)|The Magician]]''.&lt;ref&gt;{{cite web |title='Man of Will' depicts Kim Koo as young man |url=http://koreajoongangdaily.joins.com/news/article/article.aspx?aid=3039003 |website=Korea JoongAng Daily | date=28 September 2017 |accessdate=14 May 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite web |title=LEE Won-tae |url=https://www.koreanfilm.or.kr/eng/films/index/peopleView.jsp?peopleCd=20131414 |website=Korean Film Biz Zone |accessdate=14 May 2019}}&lt;/ref&gt; His second feature, ''[[The Gangster, The Cop, The Devil]]'', was invited to the [[Cannes Film Festival]] in 2019. An upcoming American remake of the film was announced in May 2019, with [[Sylvester Stallone]] and Braden Aftergood producing and [[Ma Dong-seok]] reprising his role in the remake.&lt;ref&gt;{{cite web |title=Stallone to Produce Hollywood Remake of THE GANGSTER, THE COP, THE DEVIL |url=https://www.koreanfilm.or.kr/eng/news/news.jsp?pageIndex=1&amp;blbdComCd=601006&amp;seq=5147&amp;mode=VIEW |website=Korean Film Biz Zone |accessdate=14 May 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite web |title=Sylvester Stallone to Remake South Korean Thriller 'The Gangster, The Cop, The Devil' |url=https://www.hollywoodreporter.com/news/balboa-productions-remake-south-korean-thriller-gangster-cop-devil-1207585 |website=The Hollywood Reporter |date=5 May 2019 |accessdate=14 May 2019}}&lt;/ref&gt;
3132 ** ''[[Rocky]]'', Written by [[Sylvester Stallone]]
3133 | ''[[Escape Plan: The Extractors]]'' || [[Summit Entertainment]] || [[John Herzfeld]] (director); John Herzfeld, Miles Chapman (screenplay); [[Sylvester Stallone]], [[50 Cent|Curtis Jackson]], [[Dave Bautista]] ||style=&quot;text-align:center;&quot;| &lt;ref&gt;{{Cite web |title=Escape Plan: The Extractors (2019) |url=https://www.boxofficemojo.com/movies/intl/?page=&amp;country=R2&amp;wk=2019W30&amp;id=_fESCAPEPLANTHEEXT01 |url-status=live |archive-url=https://web.archive.org/web/20230205081842/https://www.boxofficemojo.com/release/rl4177036289/weekend/ |archive-date=5 February 2023 |access-date=17 August 2019 |website=[[Box Office Mojo]]}}&lt;/ref&gt;
3134 | ''[[Rambo: Last Blood]]'' || [[Lionsgate Films|Lionsgate]] / [[Millennium Films]] || [[Adrian Grunberg]] (director); [[Sylvester Stallone]], Matt Cirulnick (screenplay); Sylvester Stallone, [[Adriana Barraza]], [[Paz Vega]], [[Yvette Monreal]], [[Sergio Peris-Mencheta]], [[Oscar Jaenada]], [[Joaquín Cosío]] ||style=&quot;text-align:center;&quot;| &lt;ref&gt;{{Cite web |title=Sylvester Stallone's 'Rambo: Last Blood' Gets September Release Date |url=https://www.thewrap.com/sylvester-stallones-rambo-last-blood-gets-september-release-date/ |url-status=live |archive-url=https://web.archive.org/web/20230205081847/https://www.thewrap.com/sylvester-stallone-rambo-last-blood-september-2019-release/ |archive-date=5 February 2023 |access-date=28 February 2019 |website=The Wrap|date=28 February 2019 }}&lt;/ref&gt;
3135 Perez's next assignment also became a very prominent fight in boxing history when, on March 24, 1975, [[Muhammad Ali vs. Chuck Wepner|Ali defended the world Heavyweight title versus Chuck Wepner]]. Perez was called in to referee this bout; his second fight refereed outside New York (the first had been Ali-Quarry I). Underdog [[Chuck Wepner|Wepner]] became an instant celebrity when, in round nine, he scored a knockdown of Ali.&lt;ref&gt;{{Cite web|url=https://boxrec.com/en/event/221|title=BoxRec: Event|website=boxrec.com}}&lt;/ref&gt; This also became a controversial matter-many have felt that Wepner only dropped Ali because Wepner stepped on Ali's foot as he landed a punch to Ali's heart. Wepner disputes this.&lt;ref&gt;{{Cite web|url=http://www.foxsports.com/boxing/story/muhammad-ali-chuck-wepner-the-bayonne-bleeder-knockdown-060316|title=Chuck Wepner calls the day he lost to Muhammad Ali the greatest of his life|first1=foxsports|last1=Jun 4|first2=2016 at 1:02a|last2=ET|date=June 4, 2016|website=FOX Sports}}&lt;/ref&gt; Nevertheless, the knockdown and fight allegedly inspired [[Sylvester Stallone]] to write and film the first [[Rocky]] film.&lt;ref name=&quot;auto&quot;&gt;{{Cite web|url=https://www.usatoday.com/story/life/movies/2017/05/02/who-is-real-rocky-chuck-wepner/101158422/|title='Chuck': Five things you should know about the real 'Rocky' Chuck Wepner|first=Bryan|last=Alexander|website=USA TODAY}}&lt;/ref&gt; Perez's refereeing that night was, ironically, unsatisfying to the winner Ali, who, according to Perez, complained that Perez let Wepner hit Ali with low blows and called him a &quot;dirty dog&quot;. Ali commented that Perez was &quot;not Black and (he's) not White-(he's) Puerto Rican but (he's) trying to be White&quot;, &quot;He is more Black than White, but he's trying to be White&quot;.
3136 The series ''Art Brothel'' consists of 25 photographs of various sizes with plastic laminate backings. The artist's intention was to create an ironic, desecrating celebration of the system of contemporary art with an &quot;overtone of parody&quot;&lt;ref&gt;Marziani, p. 50.&lt;/ref&gt; and using references to cinema. Corte created the ''Art Brothel'' series in 1995, the year of the Venice Biennale centenary, so it is not surprising that this celebration is his first ironic reference. On the backstage of an actual film set, where Corte worked as a scenographer and actor, a group of professional male, female and transgender high-class prostitutes plays in sequences that allude to well-known paintings, techniques or artistic trends chronicling the history of art. The artist himself also appears in these sequences. To create the photographic series, Corte projected the sequences on a cinema screen, and then re-photographed them. The resulting photos were later printed and cropped by the artist, sometimes leaving at their edges a small portion of the screen on which the sequences were projected. Also, French subtitles, apparently disconnected from the various scenes, have been superimposed on the photographs. The titles of the single photos are indicative of the artist's intent, but the series contains more subtle references to cinematographic art. For example, the film set is located in a showroom on [[Via Veneto]] - the famous street in Rome that was featured in the 1960 film by [[Federico Fellini]] [[La Dolce Vita]], and the show room itself belongs to an Italian architect and interior designer of villas owned by [[Sylvester Stallone]].&lt;ref&gt;Marziani, p. 50.&lt;/ref&gt;
3137 | father              = [[Sylvester Stallone]]
3138 '''Sistine Rose Stallone''' (born June 27, 1998) is an American actress, model and internet personality.&lt;ref&gt;{{Cite web|url=https://deadline.com/2018/12/47-meters-down-sequel-sistine-stallone-corinne-foxx-release-date-1202516970/|title='47 Meters Down' Sequel: Sophie Nélisse To Star, Corinne Foxx &amp; Sistine Stallone Make Film Debuts; Summer 2019 Release|last=Wiseman|first=Andreas|date=2018-12-10|website=Deadline|language=en|access-date=2019-08-03}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://theslanted.com/2019/08/34287/47-meters-down-uncaged-final-trailer/|title='47 Meters Down: Uncaged' Chomps on a Final Trailer|date=2019-08-02|website=Slanted|language=en-US|access-date=2019-08-03}}&lt;/ref&gt; She made her acting debut as Nicole in the survival horror film ''[[47 Meters Down: Uncaged]]'', directed by [[Johannes Roberts]].&lt;ref&gt;{{Cite web|url=https://www.newindianexpress.com/entertainment/english/2019/jul/26/sylvester-stallones-daughter-sistine-rose-to-debut-in-47-meters-down-sequel-2009755.html|title=Sylvester Stallone's daughter Sistine Rose to debut in '47 Meters Down' sequel|website=The New Indian Express|date=26 July 2019 |access-date=2019-08-03}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://hollywoodlife.com/2019/08/01/sistine-stallone-green-gown-thigh-high-slit-banquet-pic-photo/|title=Sistine Stallone, 22, Sizzles In Green Gown With Thigh High Leg Slit On The Red Carpet - Pics|last=Larocca|first=Courteney|date=2019-08-01|website=Hollywood Life|language=en|access-date=2019-08-03}}&lt;/ref&gt;
3139 Sistine Stallone was born in 1998 as the second daughter of actor [[Sylvester Stallone]] and former model [[Jennifer Flavin]].&lt;ref name=&quot;:0&quot; /&gt; Stallone and her sisters Sophia and Scarlet were [[Golden Globe Ambassador]]s at the [[75th Golden Globe Awards]] in January 2018.
3140 In 2016, she signed to [[IMG Models]],&lt;ref&gt;{{Cite web|url=https://www.hollywoodreporter.com/news/stallone-daughters-miss-golden-globe-gig-is-moment-960844|title=Stallone Daughters on Miss Golden Globe Gig: &quot;This Is Our Moment&quot;|website=The Hollywood Reporter|date=5 January 2017 |language=en|access-date=2019-08-12}}&lt;/ref&gt; and made her first appearance at a fashion show for [[Chanel]].&lt;ref name=&quot;:0&quot;&gt;{{Cite web|url=https://www.thisisinsider.com/sylvester-stallone-daughters-who-are-they-2017-10|title=Sylvester Stallone has 3 smart and stunning daughters - here's what you should know about Sophia, Sistine, and Scarlet|last=Oswald|first=Anjelica|website=INSIDER|date=5 October 2017 |access-date=2019-08-12}}&lt;/ref&gt; She appeared in a July 2016 issue of ''[[Glamour (magazine)|Glamour]]'' and was featured on the cover of ''[[Elle (magazine)|Elle Russia]]'' for November 2017.&lt;ref name=&quot;:0&quot; /&gt;&lt;ref&gt;{{Cite web|url=https://www.glamourmagazine.co.uk/article/meet-sistine-rose-stallone-daughter-of-sylvester-stallone|title=Meet Sylvester Stallone's daughter: A rising model star|last=Schofer|first=Simone|website=Glamour UK|date=10 June 2016 |access-date=2019-08-12}}&lt;/ref&gt;
3141 [[Category:Family of Sylvester Stallone]]
3142     <redirect title="Sylvester Stallone" />
3143       <text bytes="317" sha1="85yjxmbgcnl4rtj8ub2ipmt4iquoopz" xml:space="preserve">#REDIRECT [[Sylvester Stallone#Personal%20life]]
3144 [[Category:Sylvester Stallone]]
3145 [[Category:Family of Sylvester Stallone]]
3146     <redirect title="Sylvester Stallone" />
3147       <text bytes="95" sha1="gdeckyonjiryqm3bre9fc8upyyuub0o" xml:space="preserve">#REDIRECT [[Sylvester Stallone#Personal%20life]]
3148     <redirect title="Sylvester Stallone" />
3149       <text bytes="95" sha1="gdeckyonjiryqm3bre9fc8upyyuub0o" xml:space="preserve">#REDIRECT [[Sylvester Stallone#Personal%20life]]
3150 | ''[[Expend4bles]]'' || [[Lionsgate Films|Lionsgate]] / [[Millennium Media]] || [[Scott Waugh]] (director); [[Kurt Wimmer]], Tad Daggerhart, [[Max D. Adams|Max Adams]] (screenplay); [[Jason Statham]], [[Sylvester Stallone]], [[Megan Fox]], [[50 Cent]], [[Randy Couture]], [[Tony Jaa]], [[Iko Uwais]], [[Jacob Scipio]], [[Levy Tran]], [[Andy Garcia]], [[Dolph Lundgren]] || style=&quot;text-align:center&quot; | &lt;ref name=&quot;LionsgateOct2022&quot; /&gt;
3151 {{User:COIBot/EditSummary|id=652969500|lang=en|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=ChewNaChunkx|link=getsatisfaction.com/imdb/topics/sylvester-stallone-in-downhill-racer|sortdomain=com.getsatisfaction.|domain=getsatisfaction.com|origdiff=https://en.wikipedia.org/w/index.php?diff=914014018&amp;oldid=913518194|edit_id=-2|resolved=X|isIP=0|date=2019-09-04|time=16:50:18|wiki=en.wikipedia.org|revid=914014018|oldid=913518194|usercount=80|whitelisted=0|blacklisteduser=0|whitereason=|blackreason=|deleted=0|top=0|there=|checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=getsatisfaction.com/imdb/topics/sylvester-stallone-in-downhill-racer|domain=getsatisfaction.com|U=80|L=161|UL=2|WUL=1|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=getsatisfaction.com/imdb/topics/sylvester-stallone-in-downhill-racer|domain=getsatisfaction.com|U=80|L=161|UL=2|WUL=1|base=|basedomain=|baseip=}}}}
3152 **[[Sylvester Stallone]] in ''[[Get Carter (2000 film)|Get Carter]]''
3153 [[Jacob Scipio]] and [[Brendan Fraser]] joined the cast later in October,&lt;ref name=&quot;Scipio&quot; /&gt;&lt;ref name=&quot;Fraser&quot; /&gt; respectively as the mob boss Anthony Bressi,&lt;ref name=&quot;ScipioRole&quot; /&gt; and as the villain [[Firefly (DC Comics)|Firefly]]. Fraser previously voiced [[Robotman (Cliff Steele)|Cliff Steele / Robotman]] in the DC Comics television series
```



```
''[[Doom Patrol (TV series)|Doom Patrol]]''.&lt;ref name=&quot;Fraser&quot; /&gt; Though it was initially reported that Fraser would portray the Garfield Lynns iteration, he later clarified in July 2022 that he was playing the [[Firefly (DC Comics)#Ted Carson|Ted Carson]] iteration, with an original backstory not adapted from existing material.&lt;ref name=&quot;FraserCarson&quot; /&gt; The villain role was originally offered to [[Sylvester Stallone]], who voiced [[King Shark (DC Extended Universe)|King Shark]] in the DCEU film ''[[The Suicide Squad (film)|The Suicide Squad]]'' (2021), but &quot;things just didn't work out&quot;.&lt;ref name=&quot;FireflyStallone&quot; /&gt;&lt;ref name=&quot;StalloneSuicideSquad&quot; /&gt;
3154 &lt;ref name=&quot;FireflyStallone&quot;&gt;{{Cite tweet |number=1452719042767192064 |user=Borys_Kit |title=Sylvester Stallone first had the offer for the part but things just didn't work out. Another key role in ''Batgirl'': the son of the gangster/villain, who becomes romantically linked to...well, you can take it from there. |first=Borys |last=Kit |date=October 25, 2021 |access-date=December 30, 2021 |archive-url=https://web.archive.org/web/20211028012638/https://twitter.com/borys_kit/status/1452719042767192064 |archive-date=October 28, 2021 |url-status=live}}&lt;/ref&gt;
3155 **[[Sylvester Stallone]] for ''[[The Specialist]]''
3156 * [[Sylvester Stallone]] for ''[[Rocky II]]''
3157 | [[Sylvester Stallone]]
3158 * [[Sylvester Stallone]] in ''[[Rocky II]]''
3159 | [[Sylvester Stallone]] in ''[[Driven (2001 film)|Driven]]''
3160 | '''[[Sylvester Stallone]] in ''[[Spy Kids 3D: Game Over]]'''''
3161 |[[Sylvester Stallone]] and his multiple personalities in ''[[Spy Kids 3D: Game Over]]''
3162 *'''[[Sylvester Stallone]] for ''[[Assassins (1995 film)|Assassins]]'' and ''[[Judge Dredd (film)|Judge Dredd]]'''''
3163 ** [[Sylvester Stallone]] for ''[[Daylight (1996 film)|Daylight]]''
3164 | The fifth film in the [[Rambo (franchise)|Rambo series]], starring and co-written by [[Sylvester Stallone]], ''Rambo: Last Blood'' was released on September 20 and promptly received widespread criticism for violence and perceived racism and xenophobia in the film.
3165 | [[Sylvester Stallone]]
3166 In 1986, Kidd and his then girl-friend Alison Fisher came to New York and lived for two weeks in the [[Meatpacking District, Manhattan]], before they headed towards Tucson, Arizona, via Auto Driveway, where they bought a van in which they traveled around the USA for a year.&lt;ref&gt;{{cite web |last1=Leonardo |first1=Kathy |title=Interview: Jeremy Kidd |url=https://eattravelgo.com/stories/artist-interview-jeremy-kidd/ |website=Eat.Travel.Go |accessdate=15 February 2019}}&lt;/ref&gt; Kidd worked in migrant jobs, digging ditches and boxing mesquite trees in the Tucson desert for little money and under sometimes hottest weather conditions, alongside people with no social security cards. In 1987 he and Fisher arrived on the Westcoast. They first lived in Seattle, Washington, and then Portland, Oregon, and San Francisco, where they sold hand-painted art T-Shirts at craft fairs and galleries under the company name Fidget, [[Robin Williams]] being an early patron. Later they moved to the famed [[Topanga Canyon]] Ranch Hotel in Los Angeles where they sold their designer T-Shirts to upscale stores on [[Rodeo Drive]] in Beverly Hills to the movie stars of the time, such as by [[Silvester Stallone]], [[Steve Martin]], [[Chevy Chase]], etc. Kidd is also a surfer and surfed both in Topanga State Beach and Malibu Third Point. In 1991 he married Chevy Chase's half-sister, Catherine Cederquist, and a year later they were divorced. Kidd remarried on 3 September 2014 in Santa Barbara. His wife is Wijitra Pimros Kidd. The two met in Thailand.&lt;ref&gt;{{cite web |last1=Kidd |first1=Jeremy |title=Steckbrief: Jeremy Kidd |url=http://simonekussatz-arete.blogspot.com/2019/10/steckbrief-jeremy-kidd.html |website=ARETE | publisher=Simone Susanne Kussatz |accessdate=16 October 2019}}&lt;/ref&gt;
3167 | ''[[Guardians of the Galaxy Vol. 3]]'' || [[Marvel Studios]] || [[James Gunn]] (director/screenplay); [[Chris Pratt]], [[Zoe Saldaña]], [[Dave Bautista]], [[Karen Gillan]], [[Pom Klementieff]], [[Vin Diesel]], [[Bradley Cooper]], [[Sean Gunn]], [[Chukwudi Iwuji]], [[Will Poulter]], [[Linda Cardellini]], [[Nathan Fillion]], [[Elizabeth Debicki]], [[Maria Bakalova]], [[Sylvester Stallone]] || style=&quot;text-align:center&quot; | &lt;ref&gt;{{Cite web |last=D'Alessandro |first=Anthony |date=May 7, 2019 |title=Marvel Shows Off 'Eternals' Footage, Dates 'Ant-Man &amp; The Wasp 3', Titles 'Black Panther 2' &amp; 'Captain Marvel 2' |url=https://deadline.com/video/marvel-sizzle-reel-eternals-black-panther-ii-the-marvels/ |url-status=live |archive-url=https://web.archive.org/web/20211107062646/https://deadline.com/video/marvel-sizzle-reel-eternals-black-panther-ii-the-marvels/ |archive-date=November 7, 2021 |access-date=October 3, 2021 |website=Deadline}}&lt;/ref&gt;
3168 | [[Sylvester Stallone]]
3169 | '''[[Sylvester Stallone]]'''
3170 | '''&quot;Too Close To Paradise&quot; by [[Sylvester Stallone]]'''
3171 | Cranky Viewer's Request: Cranky Food Special - Drinks and food that are made from cranky food combinations and have similar tastes to the originals{{efn|Given:&lt;br&gt;1. Coffee milk (made from barley tea, milk and sugar)&lt;br&gt;2. Beer (made from oolong tea and carbonated water)&lt;br&gt;3. [[Misu]] (made from flour (fried in a pan), sugar and water)&lt;br&gt;4. [[Makgeolli]] (made from Morning Rice drink and Sprite)&lt;br&gt;5. Yogurt drink (made from [[Seolleongtang]], [[Naengmyeon]] soup and sugar)}}{{efn|Given:&lt;br&gt;1. Corn soup (made from pickled radish, bread and milk)&lt;br&gt;2. Salmon sushi (made from tangerine, seaweed and rice)&lt;br&gt;3. Soy sauce marinated crab (made from banana and coffee)&lt;br&gt;4. Pollack pancake (made from flour, eggs, rice and ramyeon soup powder)&lt;br&gt;5. Cheesecake (made from soft tofu and plain yogurt)}}&lt;br&gt;Cranky Neighbourhood Battle (against Don Spike, Muzie and Ravi): Park Joon-hyung's Cranky Speed Quiz&lt;br&gt;Cranky Sorting (against Don Spike, Muzie and Ravi): Best action star (based on the responses of 100 frequent cinema visitors){{efn|Given:&lt;br&gt;1. [[Sylvester Stallone]]&lt;br&gt;2. [[Chow Yun-fat]]&lt;br&gt;3. [[Jackie Chan]]&lt;br&gt;4. [[Tom Cruise]]&lt;br&gt;5. [[Robert Downey Jr.]]}}
3172 At the [[2019 Cannes Film Festival]], [[Sylvester Stallone]] revealed he had plans to reboot his 1986 action film ''[[Cobra (1986 film)|Cobra]]'' as a streaming series, say  &quot;That (conceit) was what if [[Bruce Springsteen]] had a gun? That was rock n' roll meets drama. That should have been another franchise because that character was so cool. And I blew it. My personal life got in the way. But we're trying to bring it back as a streaming TV series. Bring out the zombie squad. I'm long gone, but the idea is really good.&quot;&lt;ref&gt;{{cite web |url=https://deadline.com/2019/05/sylvester-stallone-cannes-film-festival-rambo-v-1202621832/ |title=Sylvester Stallone Celebrated At Cannes: Promises 'Serious Vengeance' In 'Rambo V', Rebooting 'Cobra' &amp; Plans For New 'Rocky' |website=[[Deadline Hollywood]] |first=Anthony |last=D'Alessandro |date=May 24, 2019 |access-date=September 19, 2020}}&lt;/ref&gt; Later that year Stallone revealed to [[Fandango Media|Fandango]] that &quot;I'm talking with Robert Rodriguez right now about Cobra, which looks like that could happen,&quot; and that &quot;It's basically his baby now.&quot;&lt;ref&gt;{{cite web |url=https://www.fandango.com/movie-news/sylvester-stallone-teaming-with-robert-rodriguez-on-cobra-reboot-is-tango-and-cash-next-753897 |title=SYLVESTER STALLONE TEAMING WITH ROBERT RODRIGUEZ ON 'COBRA' REBOOT; IS 'TANGO &amp; CASH' NEXT? |website=[[Fandango Media|Fandango]] |last=Davis |first=Erik |date=September 16, 2019 |access-date=September 30, 2020}}&lt;/ref&gt;
3173 *[[Jackie Stallone]], 98, American astrologer.&lt;ref&gt;[https://variety.com/2020/film/obituaries-people-news/jackie-stallone-dead-sylvester-stallone-mom-1234778583/ Jackie Stallone, Sylvester Stallone's Mother and Celebrity Astrologist, Dies at 98]&lt;/ref&gt;
3174 * [[Sylvester Stallone]]
3175 * Sylvester Stallone
3176 | budget         = $100 million&lt;ref&gt;{{cite web|url=https://www.thewrap.com/how-to-watch-samaritan-sylvester-stallone-movie-online-streaming-where/amp/|title=How to Watch Sylvester Stallone's Superhero Movie 'Samaritan'|website=www.thewrap.com|date=August 26, 2022|access-date=October 23, 2022|archive-date=October 23, 2022|archive-url=https://web.archive.org/web/20221023202323/https://www.thewrap.com/how-to-watch-samaritan-sylvester-stallone-movie-online-streaming-where/amp/|url-status=live}}&lt;/ref&gt;
3177 '''''Samaritan''''' is a 2022 American [[superhero film]] directed by [[Julius Avery]] and written by Bragi F. Schut. Described&lt;ref&gt;{{cite web|url=https://collider.com/samaritan-review-sylvester-stallone-prime-video/|title='Samaritan' Review: Sylvester Stallone Flexes His Muscles to Carry a Generic Superhero Story|website=www.collider.com|date=August 26, 2022|access-date=December 27, 2023}}&lt;/ref&gt; as a gritty and dark take on superhero movies, the story was previously adapted{{efn|name=original}} into the Mythos Comics graphic novels by Schut, Marc Olivent, and Renzo Podesta. It is a co-production of [[Metro-Goldwyn-Mayer Pictures]] and [[Balboa Productions]]. The film stars [[Sylvester Stallone]] in the main role, [[Javon Walton|Javon &quot;Wanna&quot; Walton]], [[Pilou Asbæk]], [[Dascha Polanco]], and [[Moisés Arias]]. The story follows a kid who suspects that his neighbor is secretly a superhero who was believed to have died many years prior.
3178 * [[Sylvester Stallone]] as Joe Smith / Samaritan / Nemesis
3179 | image2 = Sylvester Stallone Cannes 2019.jpg
3180 | footer = Director [[Julius Avery]] and lead actor/co-producer [[Sylvester Stallone]].
3181 In February 2019, it was announced that [[MGM]] had acquired a [[spec script]] by Bragi F. Schut titled ''Samaritan'' to be co-produced with [[Balboa Productions]].&lt;ref name=&quot;samaritan&quot;&gt;{{cite web|first=Dave|last=McNary|url=https://variety.com/2019/film/news/sylvester-stallone-samaritan-mgm-1203145083/|title=Sylvester Stallone's Superhero Drama 'Samaritan' Lands at MGM|date=February 21, 2019|website=[[Variety (magazine)|Variety]]|access-date=December 8, 2019|archive-date=February 22, 2019|archive-url=https://web.archive.org/web/20190222012147/https://variety.com/2019/film/news/sylvester-stallone-samaritan-mgm-1203145083/|url-status=live}}&lt;/ref&gt; Schut had written the screenplay prior to adapting the story in a series of graphic novels published by Mythos Comics.&lt;ref&gt;{{Cite tweet |last=Schut |first=Bragi |user=bragischutjr |number=1544047516441874433 | title=The script was written first. Then adapted into a comic. }}&lt;/ref&gt; In September 2019, [[Julius Avery]] joined the production as director.&lt;ref name=&quot;production&quot;&gt;{{cite web| first=Mike Jr.|last=Fleming|url=https://deadline.com/2019/09/sylvester-stallone-samaritan-julius-avery-next-film-rambo-last-blood-mgm-1202742438/|title=Rambo Retired, Sylvester Stallone To Star In 'Samaritan;' Julius Avery Set To Direct MGM Thriller|date=September 23, 2019|website=[[Deadline Hollywood]]|access-date=December 8, 2019|archive-date=September 23, 2019|archive-url=https://web.archive.org/web/20190923170340/https://deadline.com/2019/09/sylvester-stallone-samaritan-julius-avery-next-film-rambo-last-blood-mgm-1202742438/|url-status=live}}&lt;/ref&gt;
3182 In February 2019, it was announced that [[Sylvester Stallone]] would star in the titular role and also act as producer. In February 2020, [[Martin Starr]], [[Moisés Arias]], [[Dascha Polanco]], [[Pilou Asbæk]], [[Javon Walton|Javon 'Wanna' Walton]], [[Jared Odrick]], and Michael Aaron Milligan joined the cast in supporting roles.&lt;ref&gt;{{cite web |last=Carbone |first=Gin |url=https://www.cinemablend.com/news/2489866/sylvester-stallone-original-superhero-movie-samaritan-adds-5-cast-members |title=Sylvester Stallone's Original Superhero Movie Samaritan Adds 5 Cast Members |work=CinemaBlend |date=February 8, 2020 |access-date=February 9, 2020 |archive-date=February 16, 2020 |archive-url=https://web.archive.org/web/20200216161614/https://www.cinemablend.com/news/2489866/sylvester-stallone-original-superhero-movie-samaritan-adds-5-cast-members |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite web |date= February 21, 2020 |last= N'Duka |first= Amanda | title= Sylvester Stallone Thriller 'Samaritan' Adds Jared Odrick &amp; Michael Aaron Milligan; Christian DelGrosso Joins 'Wally's Wonderland' - Film Briefs |url= https://deadline.com/2020/02/sylvester-stallone-samaritan-jared-odrick-michael-aaron-milligan-christian-delgrosso-wallys-wonderland-film-briefs-1202862151/ |website= [[Deadline Hollywood]] |access-date= February 23, 2020 |archive-date= February 22, 2020 |archive-url= https://web.archive.org/web/20200222150317/https://deadline.com/2020/02/sylvester-stallone-samaritan-jared-odrick-michael-aaron-milligan-christian-delgrosso-wallys-wonderland-film-briefs-1202862151/ |url-status= live }}&lt;/ref&gt; In March 2020, [[Natacha Karam]] joined the cast of the film.&lt;ref&gt;{{cite web |url= https://www.hollywoodreporter.com/heat-vision/natacha-karam-joins-sylvester-stallone-samaritan-1282164 |title= '911: Lone Star' Actress Natacha Karam Joins Sylvester Stallone in 'Samaritan'
```

```
     (Exclusive) |website= [[The Hollywood Reporter]] |first= Borys |last= Kit |date= March 2, 2020 |access-date= March 2, 2020 |archive-date= March 2, 2020 |archive-url=
     https://web.archive.org/web/20200302233736/https://www.hollywoodreporter.com/heat-vision/natacha-karam-joins-sylvester-stallone-samaritan-1282164 |url-status= live }}&lt;/ref&gt;
3183 In September 2019, filming was scheduled for a 2020 start date in [[Atlanta]].&lt;ref name=&quot;production&quot; /&gt;&lt;ref&gt;{{cite web|title=Sylvester Stallone’s ‘Samaritan’ To Film in Atlanta|
     url=https://www.projectcasting.com/news/sylvester-stallones-samaritan-to-film-in-atlanta/|date=December 16, 2019|website=Projectcasting.com|access-date=February 3, 2020|archive-date=February 3, 2020|
     archive-url=https://web.archive.org/web/20200203180144/https://www.projectcasting.com/news/sylvester-stallones-samaritan-to-film-in-atlanta/|url-status=live}}&lt;/ref&gt; Filming was confirmed to
     have commenced by February 26, 2020.&lt;ref&gt;{{cite web|url=https://www.instagram.com/p/B8y6CU9Jl_T/  |archive-url=https://ghostarchive.org/iarchive/s/instagram/B8y6CU9Jl_T |archive-date=December
     24, 2021 |url-access=registration|title=Sly Stallone on Instagram: &quot;My daughter @sistinestallone Visiting me on the set of my new film SAMARITAN after I had to take care of business!
     #Samaritan&quot; |via=Instagram |date=2020-02-20 |access-date=2020-09-07}}{{cbignore}}&lt;/ref&gt; On March 14, the production went on hiatus due to the [[COVID-19 pandemic]].&lt;ref&gt;{{cite web|
     url=https://deadline.com/2020/03/sylvester-stallone-movie-the-samaritan-coronavirus-hiatus-1202883810/|title=Sylvester Stallone Superhero Thriller ‘Samaritan’ Going On Two-Week Hiatus|
     last=D’Alessandro|first=Anthony|work=[[Deadline Hollywood|Deadline]]|date=March 15, 2020|access-date=April 3, 2020|archive-date=March 15,
     2020|archive-url=https://web.archive.org/web/20200315194432/https://deadline.com/2020/03/sylvester-stallone-movie-the-samaritan-coronavirus-hiatus-1202883810/|url-status=live}}&lt;/ref&gt; By October
     8, 2020, filming had resumed.&lt;ref&gt;{{cite web|url=https://movieweb.com/samaritan-movie-production-restarts-sylvester-stallone-images/|title=Sylvester Stallone Shares Samaritan Sneak Peek as
     Filming Resumes|work=[[MovieWeb]]|first=Ryan|last=Scott|date=October 8, 2020|access-date=October 11, 2020|archive-date=October 10,
     2020|archive-url=https://web.archive.org/web/20201010144428/https://movieweb.com/samaritan-movie-production-restarts-sylvester-stallone-images/|url-status=live}}&lt;/ref&gt;
3184 The release of ’’Samaritan’’ was delayed several times, having been previously scheduled to be released theatrically on November 20, 2020, December 11, 2020, and June 4, 2021.&lt;ref&gt;{{cite web|
     first=Anthony|last=D’Alessandro|url=https://deadline.com/2019/11/sylvester-stallone-mgm-thriller-samaritan-opening-late-fall-2020-1202743588/|title=Sylvester Stallone MGM Thriller ‘Samaritan’ Heads
     To Early December 2020|date=November 27, 2019|website=Deadline Hollywood|access-date=December 8, 2019|archive-date=November 27, 2019|archive-url=https://web.archive.org/web/20191127213016/https://
     deadline.com/2019/11/sylvester-stallone-mgm-thriller-samaritan-opening-late-fall-2020-1202743588/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://deadline.com/2020/06/matrix-4-
     godzilla-vs-king-release-date-changes-1202958388/|title=‘Matrix 4’ Moves To 2022, ‘Godzilla Vs. Kong’ Stomps To 2021 &amp; More: Warner Bros. Release Date Change Friday|website=Deadline Hollywood|
     first=Anthony|last=D’Alessandro|date=June 12, 2020|access-date=June 12, 2020|archive-date=June 13, 2020|archive-url=https://web.archive.org/web/20200613001328/https://deadline.com/2020/06/matrix-4-
     godzilla-vs-king-release-date-changes-1202958388/|url-status=live}}&lt;/ref&gt; The film was released on August 26, 2022, in the United States by [[United Artists Releasing]] and [[Amazon Studios]],
     the latter having purchased MGM that same year, via streaming on [[Prime Video]].&lt;ref&gt;{{cite web|url=https://movieweb.com/samaritan-release-date-sylvester-stallone/|title=Sylvester Stallone
     Superhero Flick Samaritan Sets 2022 Release Date|work=Movieweb|first=Jon|last=Fuge|date=August 11, 2021|access-date=August 14, 2021|archive-date=August 11,
     2021|archive-url=https://web.archive.org/web/20210811094654/https://movieweb.com/samaritan-release-date-sylvester-stallone/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Maas |
     first=Jennifer |date=2022-03-17 |title=Amazon Closes $8.5 Billion Acquisition of MGM |url=https://variety.com/2022/tv/news/amazon-mgm-merger-close-1235207852/ |access-date=2022-03-17 |website=Variety
     |language=en-US |archive-date=April 4, 2022 |archive-url=https://web.archive.org/web/20220404000745/https://variety.com/2022/tv/news/amazon-mgm-merger-close-1235207852/ |url-status=live
     }}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/movies/movie-news/russell-crowe-popes-exorcist-1235169682/amp |title=Russell Crowe to Star in Supernatural Thriller ‘The
     Pope’s Exorcist’ (Exclusive) |website=[[The Hollywood Reporter]] |date=June 27, 2022 |access-date=June 28, 2022 |archive-date=June 27, 2022
     |archive-url=https://web.archive.org/web/20220627202840/https://www.hollywoodreporter.com/movies/movie-news/russell-crowe-popes-exorcist-1235169682/amp/ |url-status=live }}&lt;/ref&gt;&lt;ref&gt;
     {{cite web |last1=Jackson |first1=Angelique |title=Ron Howard’s ‘Thirteen Lives’ Moves to August, Pivots to Hybrid Release via MGM and Amazon (EXCLUSIVE) |url=https://variety.com/2022/film/news/ron-
     howard-thirteen-lives-amazon-prime-video-1235268658/ |website=Variety |date=17 May 2022 |access-date=July 25, 2022 |archive-date=June 19, 2022
     |archive-url=https://web.archive.org/web/20220619131002/https://variety.com/2022/film/news/ron-howard-thirteen-lives-amazon-prime-video-1235268658/ |url-status=live }}&lt;/ref&gt;
3185 {{Sylvester Stallone}}
3186 * [[Sylvester Stallone]] (2015)
3187 * [[Tom Cruise]] and [[Brad Pitt]] - ’’[[Interview with the Vampire (film)|Interview with the Vampire]]’’ / [[Sylvester Stallone]] and [[Sharon Stone]] - ’’[[The Specialist]]’’ (1994)
3188 *[[:Sylvester Stallone]] {{n}} [[User:MER-C|MER-C]] 16:47, 31 May 2020 (UTC)
3189 | ’’Lost on a Mountain in Maine’’ || [[Blue Fox Entertainment]] / [[Balboa Productions]] || Andrew Boodhoo Kightlinger (director); Luke Paradise (screenplay); Luke David Blumm, [[Paul Sparks]],
     [[Caitlin FitzGerald]], [[Ethan Slater]] || style=&quot;text-align:center&quot; |&lt;ref&gt;{{cite web|last=Bechara|first=Diego Ramos|title=Sylvester Stallone-Produced Feature ‘Lost on a Mountain In
     Maine’ Slated for November Release by Blue Fox Entertainment (Exclusive)|url=https://variety.com/2024/film/news/lost-on-a-mountain-in-maine-blue-fox-entertainment-sylvester-stallone-1236085210/|
     website=[[Variety (magazine)|Variety]]|date=July 25, 2024|access-date=July 25, 2024}}&lt;/ref&gt;
3190 Rick Nicita left Creative Artists Agency in 2008 to become [[Morgan Creek Entertainment|Morgan Creek’s]] Co-chairman and [[Chief operating officer|COO]].&lt;ref&gt;{{cite magazine|date=29 July 2008|
     title=Nicita Departs CAA for Morgan Creek|url=https://variety.com/2008/scene/markets-festivals/nicita-departs-caa-for-morgan-creek-1117989714/|magazine=Variety|access-date=15 December
     2019}}&lt;/ref&gt; Nicita departed Morgan Creek in 2011,&lt;ref&gt;{{cite magazine|date=5 January 2012|title=Rick Nicita Speaks About Morgan Creek Exit|url=https://deadline.com/2012/01/rick-nicita-
     speaks-on-his-exit-from-morgan-creek-210542/|magazine=Deadline Hollywood|access-date=15 December 2019}}&lt;/ref&gt;&lt;ref&gt;{{cite news|date=5 January 2012|title=Rick Nicita Exiting Morgan Creek
     (Exclusive)|url=https://www.hollywoodreporter.com/news/rick-nicita-morgan-creek-278797|publisher=The Hollywood Reporter|access-date=15 December 2019}}&lt;/ref&gt; later returning to talent
     representation&lt;ref&gt;{{cite news|date=18 December 2015|title=Sylvester Stallone Signs With Manager Rick Nicita|url=https://sg.style.yahoo.com/sylvester-stallone-signs-manager-rick-204118552.html|
     publisher=[[Yahoo!|Yahoo! Lifestyle]]|access-date=15 December 2019}}&lt;/ref&gt; via his self-created RPMedia shingle. (RP is an [[acronym]] for &quot;Rick and Paula&quot;).&lt;ref&gt;{{cite news|
     date=1 November 2012|title=Hollywood Player Joins the Club on Broadway|url=https://www.nytimes.com/2012/11/04/theater/paula-wagner-turns-to-producing-on-broadway.html|url-access=subscription|
     newspaper=The New York Times|access-date=15 December 2019}}&lt;/ref&gt;
3191 {{User:COIBot/EditSummary|id=348800140|lang=fr|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Jean-Jacques Georges|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-years-
     ago-allegedly-group-sex-teen-police-say/2016/02/16|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://fr.wikipedia.org/w/index.php?
     diff=142865554&amp;oldid=142865223&amp;rcid=348596073|edit_id=-2|resolved=X|isIP=0|date=2017-11-23|time=12:45:41|wiki=fr.wikipedia.org|revid=142865554|oldid=142865223|usercount=8434|whitelisted=1|
     blacklisteduser=0|whitereason=User is in a trusted group reviewer on some wikis|blackreason=|deleted=0|top=0|there=|checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=time.com/5029300/
     sylvester-stallone-sexual-assault-allegations/|domain=time.com|U=8434|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-
     years-ago-allegedly-group-sex-teen-police-say/2016/02/16|domain=baltimorepostexaminer.com|U=8434|L=213|UL=X|WUL=X|base=|basedomain=|baseip=}}}}
3192 {{User:COIBot/EditSummary|id=357659550|lang=en|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Posters5|link=baltimorepostexaminer.com/ask-sylvester-stallone-40-year-old-man-group-sex-
     masturbate-front-minor/2017/11/10|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://en.wikipedia.org/w/index.php?diff=814323702&amp;oldid=813952376|edit_id=-2|
     resolved=X|isIP=0|date=2017-12-08|time=03:42:58|wiki=en.wikipedia.org|revid=814323702|oldid=813952376|usercount=91|whitelisted=0|blacklisteduser=0|whitereason=|blackreason=|deleted=0|top=0|there=|
     checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-years-ago-allegedly-group-sex-teen-police-say/2016/02/16|
     domain=baltimorepostexaminer.com|U=91|L=213|UL=4|WUL=1|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-years-ago-allegedly-group-sex-
     teen-police-say/2016/02/16|domain=baltimorepostexaminer.com|U=91|L=213|UL=4|WUL=1|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=baltimorepostexaminer.com/ask-sylvester-stallone-40-year-
     old-man-group-sex-masturbate-front-minor/2017/11/10|domain=baltimorepostexaminer.com|U=91|L=213|UL=4|WUL=1|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=baltimorepostexaminer.com/ask-
     sylvester-stallone-40-year-old-man-group-sex-masturbate-front-minor/2017/11/10|domain=baltimorepostexaminer.com|U=91|L=213|UL=4|WUL=1|base=|basedomain=|baseip=}}&amp;#32;’’Additional links removed
     for [[Wikipedia:Template limits#Post-expand include size|improved performance]]. See [//en.wikipedia.org/w/index.php?title=Wikipedia:WikiProject_Spam/LinkReports/
     baltimorepostexaminer.com&amp;action=history page history].’’}}
3193 {{User:COIBot/EditSummary|id=411904448|lang=en|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Abbyjjjj96|link=baltimorepostexaminer.com/ask-sylvester-stallone-40-year-old-man-group-sex-
     masturbate-front-minor/2017/11/10|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://en.wikipedia.org/w/index.php?diff=824167994&amp;oldid=823792096|edit_id=-2|
     resolved=X|isIP=0|date=2018-02-05|time=18:42:16|wiki=en.wikipedia.org|revid=824167994|oldid=823792096|usercount=195|whitelisted=0|blacklisteduser=0|whitereason=|blackreason=|deleted=0|top=0|there=|
     checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=www.dailymail.co.uk/news/article-5199015/Sylvester-Stallone-denies-second-rape-accusers-claims.html|domain=dailymail.co.uk|U=195|L=-1|
     UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-years-ago-allegedly-group-sex-teen-police-say/2016/02/16|
     domain=baltimorepostexaminer.com|U=195|L=213|UL=2|WUL=1|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=baltimorepostexaminer.com/ask-sylvester-stallone-40-year-old-man-group-sex-masturbate-
     front-minor/2017/11/10|domain=baltimorepostexaminer.com|U=195|L=213|UL=2|WUL=1|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.dailymail.co.uk/news/article-5081605/Sylvester-Stallone-
     accused-forcing-teen-threesome.html|domain=dailymail.co.uk|U=195|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}}}
3194 {{User:COIBot/EditSummary|id=479688777|lang=en|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Thresholy|link=baltimorepostexaminer.com/ask-sylvester-stallone-40-year-old-man-group-sex-
     masturbate-front-minor/2017/11/10|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://en.wikipedia.org/w/index.php?diff=845913833&amp;oldid=845802679|edit_id=-2|
     resolved=X|isIP=0|date=2018-06-15|time=00:22:54|wiki=en.wikipedia.org|revid=845913833|oldid=845802679|usercount=22|whitelisted=0|blacklisteduser=0|whitereason=|blackreason=|deleted=0|top=0|there=|
     checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=www.usatoday.com/story/life/2018/06/14/sylvester-stallone-sex-crime-allegation-under-review-prosecutors/701542002/|domain=usatoday.com|
     U=22|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.usatoday.com/story/life/2018/06/14/sylvester-stallone-sex-crime-allegation-under-review-prosecutors/701542002/|
     domain=usatoday.com|U=22|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.usatoday.com/story/life/people/2017/11/16/rep-sexual-assault-claim-against-sylvester-stallone-
     categorically-false/872915001/|domain=usatoday.com|U=22|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.usatoday.com/story/life/people/2017/11/16/rep-sexual-assault-
     claim-against-sylvester-stallone-categorically-false/872915001/|domain=usatoday.com|U=22|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}&amp;#32;’’Additional links removed for [[Wikipedia:Template
     limits#Post-expand include size|improved performance]]. See [//en.wikipedia.org/w/index.php?title=Wikipedia:WikiProject_Spam/LinkReports/baltimorepostexaminer.com&amp;action=history page
     history].’’}}
3195 {{User:COIBot/EditSummary|id=575951869|lang=fr|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Kelam|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-years-ago-allegedly-
     group-sex-teen-police-say/2016/02/16|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://fr.wikipedia.org/w/index.php?
     diff=154853548&amp;oldid=154851737&amp;rcid=377265311|edit_id=-2|resolved=X|isIP=0|date=2018-12-16|time=13:05:14|wiki=fr.wikipedia.org|revid=154853548|oldid=154851737|usercount=26999|whitelisted=1|
     blacklisteduser=0|whitereason=User is in a trusted group sysop on some wikis|blackreason=|deleted=0|top=0|there=|checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=www.lci.fr/
     international/agression-sexuelle-le-nom-de-sylvester-stallone-figurerait-dans-un-rapport-de-police-2070657.html|domain=lci.fr|U=26999|L=4407|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/
     OtherLinks|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-years-ago-allegedly-group-sex-teen-police-say/2016/02/16|domain=baltimorepostexaminer.com|U=26999|L=213|UL=X|WUL=X|base=|
```

```
basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.gala.fr/l_actu/news_de_stars/sylvester_stallone_aurait_abuse_de_sa_demi-soeur_281035|domain=gala.fr|U=26999|L=4392|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.20minutes.fr/cinema/2193179-20171225-sylvester-stallone-vise-nouvelle-plainte-agression-sexuelle|domain=20minutes.fr|U=26999|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}&amp;#32;''Additional links removed for [[Wikipedia:Template limits#Post-expand include size|improved performance]]. See [//en.wikipedia.org/w/index.php?title=Wikipedia:WikiProject_Spam/LinkReports/baltimorepostexaminer.com&amp;action=history page history].''}}
3196 {{User:COIBot/EditSummary|id=581650821|lang=fr|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Salebot|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-years-ago-allegedly-group-sex-teen-police-say/2016/02/16|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://fr.wikipedia.org/w/index.php?diff=155681392&amp;oldid=155681390&amp;rcid=378880153|edit_id=-2|resolved=X|isIP=0|date=2019-01-10|time=13:51:37|wiki=fr.wikipedia.org|revid=155681392|oldid=155681390|usercount=609700|whitelisted=1|blacklisteduser=0|whitereason=User is in a trusted group bot on some wikis|blackreason=|deleted=0|top=0|there=|checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=www.allocine.fr/personne/fichepersonne_gen_cpersonne=5311.html?nopub=1|domain=allocine.fr|U=609700|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.imdb.com/name/nm0000230/|domain=imdb.com|U=609700|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.sylvesterstallone.com|domain=sylvesterstallone.com|U=609700|L=|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.worldcat.org/identities/lccn-n77-002638|domain=worldcat.org|U=609700|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}&amp;#32;''Additional links removed for [[Wikipedia:Template limits#Post-expand include size|improved performance]]. See [//en.wikipedia.org/w/index.php?title=Wikipedia:WikiProject_Spam/LinkReports/baltimorepostexaminer.com&amp;action=history page history].''}}
3197 {{User:COIBot/EditSummary|id=646821794|lang=tr|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Totenkopf division|link=baltimorepostexaminer.com/ask-sylvester-stallone-40-year-old-man-group-sex-masturbate-front-minor/2017/11/10|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://tr.wikipedia.org/w/index.php?diff=20912795&amp;oldid=20912768|edit_id=-2|resolved=X|isIP=0|date=2019-08-24|time=13:24:53|wiki=tr.wikipedia.org|revid=20912795|oldid=20912768|usercount=13650|whitelisted=0|blacklisteduser=0|whitereason=|blackreason=|deleted=0|top=0|there=|checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=www.thedailybeast.com/articles/2014/08/14/rambo-hates-guns-how-sylvester-stallone-became-the-most-anti-gun-celeb-in-hollywood.html|domain=thedailybeast.com|U=13650|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.vanityfair.com/hollywood/2016/12/sylvester-stallone-trump-arts|domain=vanityfair.com|U=13650|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=blogs.telegraph.co.uk/news/timstanley/100127506/like-rocky-balboa-rick-santorum-is-a-working-class-hero/|domain=blogs.telegraph.co.uk|U=13650|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=variety.com/2016/film/news/sylvester-stallone-donald-trump-1201675595/|domain=variety.com|U=13650|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}&amp;#32;''Additional links removed for [[Wikipedia:Template limits#Post-expand include size|improved performance]]. See [//en.wikipedia.org/w/index.php?title=Wikipedia:WikiProject_Spam/LinkReports/baltimorepostexaminer.com&amp;action=history page history].''}}
3198 {{User:COIBot/EditSummary|id=661974610|lang=fr|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Kelam|link=baltimorepostexaminer.com/sylvester-stallone-accused-30-years-ago-allegedly-group-sex-teen-police-say/2016/02/16|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://fr.wikipedia.org/w/index.php?diff=162770212&amp;oldid=162766303&amp;rcid=394312756|edit_id=-2|resolved=X|isIP=0|date=2019-09-18|time=07:50:57|wiki=fr.wikipedia.org|revid=162770212|oldid=162766303|usercount=26999|whitelisted=1|blacklisteduser=0|whitereason=User is in a trusted group sysop on some wikis|blackreason=|deleted=0|top=0|there=|checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=www.20minutes.fr/cinema/2193179-20171225-sylvester-stallone-vise-nouvelle-plainte-agression-sexuelle|domain=20minutes.fr|U=26999|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.gala.fr/l_actu/news_de_stars/sylvester-stallone-accuse-dagression-sexuelle-son-ex-femme-brigitte-nielsen-le-defend_409452|domain=gala.fr|U=26999|L=4392|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=time.com/5029300/sylvester-stallone-sexual-assault-allegations/|domain=time.com|U=26999|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.lci.fr/international/agression-sexuelle-le-nom-de-sylvester-stallone-figurerait-dans-un-rapport-de-police-2070657.html|domain=lci.fr|U=26999|L=4407|UL=X|WUL=X|base=|basedomain=|baseip=}}&amp;#32;''Additional links removed for [[Wikipedia:Template limits#Post-expand include size|improved performance]]. See [//en.wikipedia.org/w/index.php?title=Wikipedia:WikiProject_Spam/LinkReports/baltimorepostexaminer.com&amp;action=history page history].''}}
3199 {{User:COIBot/EditSummary|id=724114672|lang=diq|wikidomain=w|namespace=|pagename=Sylvester Stallone|username=Xorasan|link=baltimorepostexaminer.com/ask-sylvester-stallone-40-year-old-man-group-sex-masturbate-front-minor/2017/11/10|sortdomain=com.baltimorepostexaminer.|domain=baltimorepostexaminer.com|origdiff=https://diq.wikipedia.org/w/index.php?diff=399157&amp;oldid=396564|edit_id=-2|resolved=X|isIP=0|date=2019-11-04|time=12:50:34|wiki=diq.wikipedia.org|revid=399157|oldid=396564|usercount=1689|whitelisted=0|blacklisteduser=0|whitereason=|blackreason=|deleted=0|top=0|there=|checked=0|coiflag=0|otherlinks={{User:COIBot/OtherLinks|link=web.archive.org/web/20100820014415/http://www.catholic.org/national/national_story.php?id=22474|domain=web.archive.org|U=1689|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.tmz.com/2018/10/30/sylvester-stallone-no-charges-sexual-assault-rape-accuser/|domain=tmz.com|U=1689|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.chedonna.it/2018/10/31/accusato-di-violenza-sessuale-sylvester-stallone-non-sara-processato/|domain=chedonna.it|U=1689|L=76|UL=X|WUL=X|base=|basedomain=|baseip=}}{{User:COIBot/OtherLinks|link=www.usatoday.com/story/life/2018/06/14/sylvester-stallone-sex-crime-allegation-under-review-prosecutors/701542002/|domain=usatoday.com|U=1689|L=-1|UL=X|WUL=X|base=|basedomain=|baseip=}}&amp;#32;''Additional links removed for [[Wikipedia:Template limits#Post-expand include size|improved performance]]. See [//en.wikipedia.org/w/index.php?title=Wikipedia:WikiProject_Spam/LinkReports/baltimorepostexaminer.com&amp;action=history page history].''}}
3200 **Woodroar, the editor in question is one user from Bulgaria who has been bent on removing information for the sake of removing it, even if the entry has the full disclosure of &quot;archive footage&quot; for the Men in Black entry in Sylvester Stallone's filmography page. This user has been known to revert this at least several times in 2019, so no, I am not going to apologize for having a short fuse with that particular user.  In the case of the Halo characters, the character I removed only appeared in one novel and was never even implemented into any of the proper game entries whose only purpose was to be cannon-fodder.  I thought the idea of that particular list of characters is if they appeared in &quot;multiple mediums&quot; or primarily came from the games. [[User:Dibol|Dibol]] ([[User talk:Dibol|talk]]) 03:18, 31 January 2020 (UTC)
3201 '''Comment''': {{s|for the record, I'm not from Bulgaria and I've never edited [[Sylvester Stallone filmography]] before. I'm not sure if this is a [[WP:CIR|CIR]] issue or simply Dibol seeing enemies wherever they go, which points to their battleground mentality.}} Note that Dibol is ''still'' [https://en.wikipedia.org/w/index.php?title=Sylvester_Stallone_filmography&amp;diff=prev&amp;oldid=938458564 calling good-faith edits &quot;vandalizing&quot;] and they're up to 3RR on that page (4RR in under 26 hours) without ever joining the Talk page discussion. [[User:Woodroar|Woodroar]] ([[User talk:Woodroar|talk]]) 22:34, 31 January 2020 (UTC)
3202 **Woodroar, I did NOT accuse you of editing the Sylvester Stallone page, for the record. I was addressing YOUR QUESTION about the IP user in question. I did the geo-locator on the the anonymous IP in question, and it has been the same user reverting the data without rhyme or reason, despite the fact that archive footage of the actor was used in Men in Black. Less confused now? Good. As for me not bothering with the conversation. The IP user in question has a very poor level of reading comprehension, and it's obvious that particular IP user in question is not a native English user. [[User:Dibol|Dibol]] ([[User talk:Dibol|talk]]) 05:22, 1 February 2020 (UTC)
3203 |7/4/2022||7/24/2022||&quot;First Look at Fall TV,&quot; featuring [[Sylvester Stallone]] of ''[[Tulsa King]]''||Photograph
3204 In 2008, Williams began producing and starring in the independent films, include the leading roles in ''Skeletons in the Desert'' (2008), ''Raven'' (2010), and ''[[Mysteria (film)|Mysteria]]'' (2011). She produced and co-starred opposite [[Samantha Morton]] in the 2013 horror film ''[[The Harvest (2013 film)|The Harvest]]'', and well as action film ''[[Den of Thieves (film)|Den of Thieves]]'' (2018) and romantic drama ''[[After (2019 film)|After]]'' (2019).&lt;ref name=dj&gt;{{Cite web|url=http://www.digitaljournal.com/entertainment/entertainment/meadow-williams-talks-after-film-motivations-and-acting-advice/article/547364|title=Meadow Williams talks 'After' film, motivations and acting advice (Includes interview)|date=April 10, 2019|website=www.digitaljournal.com}}&lt;/ref&gt; In 2017, she won [[Daytime Emmy Award for Outstanding Digital Daytime Drama Series]] as a producer of web-series ''[[The Bay (web series)|The Bay]]''.&lt;ref name=&quot;NATAS 2017 Wins&quot;/&gt; She starred opposite [[Sylvester Stallone]] in the 2018 action film ''[[Backtrace (film)|Backtrace]]'',&lt;ref&gt;{{Cite web|url=https://www.rottentomatoes.com/m/backtrace|title=Backtrace (2018)|via=www.rottentomatoes.com}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://variety.com/2018/film/news/backtrace-review-1203089356/|title=Film Review: 'Backtrace'|first1=Joe|last1=Leydon|date=December 14, 2018}}&lt;/ref&gt; and alongside [[Bruce Willis]] in the 2019 thriller ''[[10 Minutes Gone]]''.&lt;ref&gt;{{Cite web|url=https://www.hollywoodreporter.com/review/10-minutes-gone-1243578|title='10 Minutes Gone': Film Review|website=The Hollywood Reporter}}&lt;/ref&gt; She next played the role of [[Mildred Gillars]] in the drama film ''[[American Traitor: The Trial of Axis Sally]]'' opposite [[Al Pacino]].&lt;ref&gt;{{Cite web|url=https://deadline.com/2018/11/al-pacino-axis-sally-nazi-propaganda-voice-sally-gillars-michael-polish-moviepass-films-meadow-williams-1202488517/|title=Al Pacino, Meadow Williams Star In MoviePass Films Drama 'Axis Sally'|first1=Mike Jr. |last1=Fleming|date=November 1, 2018}}&lt;/ref&gt;
3205 A {{cvt|82.9|cm|in}} high version, cast by Rudier in 1902-5, one of just seven known castings made during Rodin's lifetime, was sold for £11.57m at [[Sotheby's]] in 2016.  It was sold from the collection of [[Sylvester Stallone]], who had nicknamed it the &quot;flying beaver&quot;. Stallone had acquired it at auction in 2007 for £4.6m, then a record for Rodin. [[Lucian Freud]] also owned a copy for many years, which he placed at the end of his bed, making it the first thing he saw in the morning.&lt;ref&gt;{{Cite news |last=Brown |first=Mark |last2=correspondent |first2=Mark Brown Arts |date=2016-01-27 |title=Rodin's explicit Iris and daring Freud nude go on sale for estimated £13m-plus |language=en-GB |work=The Guardian |url=https://www.theguardian.com/artanddesign/2016/jan/27/rodins-explicit-iris-daring-freud-nude-sale-combined-estimate-over-13m |access-date=2023-03-09 |issn=0261-3077}}&lt;/ref&gt;
3206 | ''[[Samaritan (film)|Samaritan]]'' || [[Amazon Studios]] / [[Metro-Goldwyn-Mayer]] / [[Balboa Productions]] || [[Julius Avery]] (director); Bragi F. Schut (screenplay); [[Sylvester Stallone]], [[Javon Walton]], [[Pilou Asbæk]], [[Dascha Polanco]], [[Moisés Arias]] ||style=&quot;&quot;| &lt;ref&gt;{{Cite web |last=Fuge |first=Jon |date=August 11, 2021 |title=Sylvester Stallone Superhero Flick Samaritan Sets 2022 Release Date |url=https://movieweb.com/samaritan-release-date-sylvester-stallone/ |url-status=live |archive-url=https://web.archive.org/web/20210811094654/https://movieweb.com/samaritan-release-date-sylvester-stallone/ |archive-date=August 11, 2021 |access-date=October 5, 2021 |website=Movieweb}}&lt;/ref&gt;
3207 | ''[[Goodnight Mommy (2022 film)|Goodnight Mommy]]'' || [[Amazon Studios]] || Matt Sobel (director); Kyle Warren (screenplay); [[Naomi Watts]], [[Cameron Crovetti]], Nicholas Crovetti, [[Peter Hermann (actor)|Peter Hermann]], Crystal Lucas-Perry, [[Jeremy Bobb]] ||style=&quot;&quot;|&lt;ref name=&quot;Amazon&quot;&gt;{{Cite web |last=D'Alessandro |first=Anthony |date=August 1, 2022 |title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan' &amp; More |url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/ |url-status=live |archive-url=https://web.archive.org/web/20220811110400/https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/ |archive-date=August 11, 2022 |access-date=August 1, 2022 |website=Deadline Hollywood |language=en-US}}&lt;/ref&gt;
3208 | '''''[[Samaritan (2022 film)|Samaritan]]''''' || [[Julius Avery]] || [[Sylvester Stallone]], [[Martin Starr]], [[Moisés Arias]], [[Dascha Polanco]] || United States || Superhero thriller&lt;ref&gt;{{cite web|first=Jon|last=Fuge|title=Sylvester Stallone Superhero Flick Samaritan Sets 2022 Release Date|url=https://movieweb.com/samaritan-release-date-sylvester-stallone/|date=August 11, 2021|website=Movie Web|access-date=October 21, 2021}}&lt;/ref&gt;
3209 | '''''[[Expend4bles]]''''' || [[Scott Waugh]] || [[Jason Statham]], [[Sylvester Stallone]], [[Dolph Lundgren]], [[Randy Couture]], [[50 Cent|Curtis &quot;50 Cent&quot; Jackson]], [[Megan Fox]], [[Tony Jaa]], [[Iko Uwais]], [[Jacob Scipio]], [[Levy Tran]], [[Andy García]] || United States || Action&lt;ref&gt;{{cite web|title='Expend4bles': New cast members, more explosions and a nonsensical plot|url=https://www.washingtonpost.com/style/2023/09/21/expendables-4-movie-review/|website=[[The Washington Post]]|date=21 September 2021|access-date=29 January 2024|first=Mark|last=Jenkins|archive-
```



```
date=29 January 2024|archive-url=https://web.archive.org/web/20240129105208/https://www.washingtonpost.com/style/2023/09/21/expendables-4-movie-review/}}&lt;/ref&gt;
| '''''[[Guardians of the Galaxy Vol. 3]]''''' || [[James Gunn]] || [[Chris Pratt]], [[Zoe Saldaña]], [[Dave Bautista]], [[Vin Diesel]], [[Bradley Cooper]], [[Karen Gillan]], [[Pom Klementieff]], [[Sean Gunn]], [[Sylvester Stallone]], [[Chukwudi Iwuji]], [[Will Poulter]] || United States ||Superhero, comedy&lt;ref&gt;{{cite web|first=Britta|last=Devore|title=Sylvester Stallone Confirms His Return to the MCU in 'Guardians of the Galaxy Vol. 3'|url=https://collider.com/guardians-of-the-galaxy-3-sylvester-stallone/|date=November 12, 2021|publisher=Collider|access-date=February 22, 2022}}&lt;/ref&gt;
| '''''[[Levon's Trade]]''''' || [[David Ayer]] || [[Jason Statham]], [[David Harbour]], [[Michael Peña]], [[Jason Flemyng]], Arianna Rivas || United States ||Action thriller&lt;ref&gt;{{cite web|last=Wiseman|first=Andreas|title=Michael Peña, David Harbour, Jason Flemyng &amp; Arianna Rivas Join Jason Statham In David Ayer's 'Levon's Trade'; Amazon MGM Sets Release Date For Black Bear Action Pic Scripted By Sylvester Stallone|url=https://deadline.com/2024/04/david-ayer-jason-statham-slyvester-stallone-movie-levons-trade-begins-amazon-mgm-release-date-1235878520/|date=April 8, 2024|website=Deadline Hollywood|access-date=June 12, 2024}}&lt;/ref&gt;
| '''''[[Demolition Man#Sequel|Demolition Man 2]]''''' || TBA || [[Slyvester Stallone]] || United States ||Action sci-fi comedy&lt;ref&gt;{{cite web|first=Clark|last=Collis|title=Sylvester Stallone is 'working on' a sequel to Demolition Man|url=https://ew.com/movies/demolition-man-sequel-sylvester-stallone/|date=May 4, 2020|website=Entertainment Weekly|access-date=June 11, 2021}}&lt;/ref&gt;
|[[Sylvester Stallone]] [[Jason Statham]] [[Dolph Lundgren]] [[Randy Couture]] [[Costas Mandylor]] [[Zoe Saldaña]] [[Jude Law]] [[Ji Chang-wook]] [[Don Wilson (kickboxer)|Don &quot;The Dragon&quot; Wilson]] [[Bai Ling]] [[Mark Dacascos]] [[Kelly Hu]] Hector David Jr [[Akihiro Kitamura]]
| style=&quot;text-align:center;&quot; |&lt;ref&gt;{{Cite web |title=Sylvester Stallone's 'Guardians of the Galaxy Vol. 2' character has a big future in the MCU |url=https://torontosun.com/2017/04/25/sylvester-stallones-guardians-of-the-galaxy-vol-2-character-has-a-big-future-in-the-mcu |access-date=2023-01-22 |website=thewhig |language=en-CA}}&lt;/ref&gt;
In 1996, stuntman and actor Matt McColm has been labelled by ''[[Los Angeles Times]]'' as &quot;Hollywood's next-generation action star,&quot; following in the footsteps of [[Arnold Schwarzenegger]] and [[Sylvester Stallone]].&lt;ref&gt;{{Cite web|url=https://news.google.com/newspapers?nid=1454&amp;dat=19961130&amp;id=FWh5AAAAIBAJ&amp;pg=6818,6314399|title=MOVIE TRENDS / What's happened to all the shoot-'em-up films? Who's the next Sly or Jean-Claude?|work=[[Los Angeles Times]]|access-date=March 23, 2020}}&lt;/ref&gt; His previous entries in the action genre included ''[[Red Scorpion 2]]'' (1994) and ''[[Subterfuge (1996 film)|Subterfuge]]'' (1996). ''The Protector'' gave McColm a chance to impress the audiences with both his perfect musculature and martial-arts skills-since he has a black belt in [[American Kenpo|Kenpo Karate]].&lt;ref&gt;{{Cite web|url=https://ultimateactionmovies.com/matt-mccolm-subterfuge-1996/|title=MATT MCCOLM IN THE CANNON FILMS-WORTHY 'SUBTERFUGE' (1996)|publisher=Ultimate Action Movies|access-date=March 23, 2020}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://action-flix.com/2019/09/30/action-rewind-matt-mccolm-delivers-the-fisticuffs-and-firepower-in-the-90s-action-gem-body-armor/|archive-url=https://archive.today/20200316173729/https://action-flix.com/2019/09/30/action-rewind-matt-mccolm-delivers-the-fisticuffs-and-firepower-in-the-90s-action-gem-body-armor/|url-status=dead|archive-date=March 16, 2020|title=ACTION REWIND: Matt McColm Delivers the Fisticuffs and Firepower in the 90's Action Gem BODY ARMOR!|publisher=Action-Flix|access-date=March 22, 2020}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://news.google.com/newspapers?nid=1309&amp;dat=19920619&amp;id=6l1PAAAAIBAJ&amp;pg=4501,3806262|title=Movie-type who shuns partying|publisher=New Straits Times|access-date=March 22, 2020}}&lt;/ref&gt;&lt;ref name=&quot;dam&quot;&gt;{{Cite web|url=http://viendammage.blogspot.com/2014/01/crappy-action-movie-craption-body-armor.html|archive-url=https://archive.today/20200322115514/http://viendammage.blogspot.com/2014/01/crappy-action-movie-craption-body-armor.html|url-status=dead|archive-date=March 22, 2020|title=Crappy + Action Movie = Craption! Body Armor|publisher=Dammaged Goods|access-date=March 23, 2020}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=http://www.the-unknown-movies.com/unknownmovies/reviews/rev150.html|title=Body Armor - The Unknown Movies|publisher=The Unknown Movies|access-date=March 23, 2020}}&lt;/ref&gt; Prior to the filming director Jack Gill worked primarily as a stuntman, stunt coordinator, and second unit director.&lt;ref name=&quot;dam&quot;/&gt;
    <title>Category:Video games based on works by Sylvester Stallone</title>
After getting clean with the help of a rehab programme,  Moore took a trip to [[Thailand]] as a means to turn his life around, hoping to give up drugs, alcohol and burglary, and start fresh as a [[Boxing|boxer]] and [[Stunt performer|stunt man]]. He arrived in Thailand in 2005 and taught English there. While he was clean, he even worked as a [[stunt double]] for [[Sylvester Stallone]] on ''[[Rambo (2008 film)|Rambo IV]]''. It was when he got back into fighting there that he became involved with drugs and crime. He had started to train in [[Muay Thai|Muay Thai boxing]], the country's national sport. &quot;I got involved with underground fighting and found bad company again.&quot; He became addicted to [[Methamphetamine|crystal meth]] and [[ya ba]] (a highly addictive methamphetamine).&lt;ref&gt;{{Cite web |title=A Prayer Before Dawn vs. the True Story of Boxer Billy Moore |url=https://www.historyvshollywood.com/reelfaces/a-prayer-before-dawn/ |access-date=2022-02-27 |website=HistoryvsHollywood.com}}&lt;/ref&gt;
| The [[Justice League]] wasn't the only DCEU property that saw redemption this year. In 2016, [[David Ayer]] made ''Suicide Squad'', and it was terrible. Jump to 2021, when [[James Gunn]] (director of Marvel's ''[[Guardians of the Galaxy (film)|Guardians of the Galaxy]]'' who jumped ship to DC after Disney temporarily fired him for some unsavory old tweets) made '''''The''' Suicide Squad'', a sequel/reboot/some weird combination of the two... and it was amazing! ''The Suicide Squad'' retains some [[Suicide Squad]] members from ''Suicide Squad'' (of course, Warner Bros. would never get rid of [[Margot Robbie]]'s [[Harley Quinn]]) but mostly follows a new set of characters including [[Idris Elba]]'s [[Bloodsport (character)|Bloodsport]], [[Sylvester Stallone]]'s [[King Shark]], and [[Daniela Melchior]]'s [[Ratcatcher (comics)|Ratcatcher]] as they attempt to destroy [[Starro the Conquerer]], a giant alien starfish.
| based_on       = {{Based on|[[List of Rocky characters|Characters]]|[[Sylvester Stallone]]}}
* Sylvester Stallone
'''''Creed III''''' is a 2023 American [[Sports film|sports]] [[Drama (film and television)|drama film]] starring and directed by [[Michael B. Jordan]] in his directional debut, and produced by Jordan, [[Irwin Winkler]], [[Charles Winkler]], William Chartoff, David Winkler, [[Ryan Coogler]], Elizabeth Raposo, [[Jonathan Glickman]], and [[Sylvester Stallone]]. It is the sequel to ''[[Creed II]]'' (2018), the third installment in the [[Creed (film series)|''Creed'' film series]], and the ninth overall in the [[Rocky (film series)|''Rocky'' film series]]. The film was written by Keenan Coogler and [[Zach Baylin]] from a story they co-wrote with Ryan Coogler. The film sees the accomplished and recently retired boxer [[Adonis Creed]] (portrayed by Jordan) come face-to-face with his childhood friend and former boxing prodigy Damian Anderson ([[Jonathan Majors]]). [[Tessa Thompson]], [[Jonathan Majors]], [[Wood Harris]], Mila Davis-Kent, [[Florian Munteanu]] and [[Phylicia Rashad]] also star.
A third ''Creed'' film was officially announced in September 2019, alongside Jordan's return; he was also confirmed to be making his directorial debut in October 2020. Majors and the remainder of the cast joined between November 2021 and September 2022; ''Creed III'' is the first film in the series not to feature Sylvester Stallone reprising his role as [[Rocky Balboa]], though he is credited as a producer. [[Principal photography]] began in January 2022 and lasted until that April, with filming locations including [[Los Angeles]], [[Tampa, Florida|Tampa]], and [[Georgia (U.S. state)|Georgia]].
In December 2018, in response to the suggestion that [[Deontay Wilder]] could play the son of [[Clubber Lang]] in a potential sequel to ''[[Creed II]]'' (2018), Sylvester Stallone and Michael B. Jordan expressed interest.&lt;ref&gt;{{Cite web |date=December 26, 2018 |title=Michael B. Jordan Clubber Lang's Son in ''Creed 3''? ... I Like It! |url=https://www.tmz.com/2018/12/26/michael-b-jordan-creed-3-deontay-wilder-clubber-lang/ |url-status=live |archive-url=https://web.archive.org/web/20201028220049/https://www.tmz.com/2018/12/26/michael-b-jordan-creed-3-deontay-wilder-clubber-lang/ |archive-date=October 28, 2020 |access-date=January 16, 2021 |website=[[TMZ Sports]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Benjamin |first=Cody |date=December 13, 2018 |title=Sylvester Stallone says Deontay Wilder could play Clubber Lang's son in ''Creed III'' |url=https://www.cbssports.com/boxing/news/sylvester-stallone-says-deontay-wilder-could-play-clubber-langs-son-in-creed-iii/ |url-status=live |archive-url=https://web.archive.org/web/20200518211828/https://www.cbssports.com/boxing/news/sylvester-stallone-says-deontay-wilder-could-play-clubber-langs-son-in-creed-iii/ |archive-date=May 18, 2020 |access-date=April 12, 2020 |website=[[CBS Sports]] |publisher=[[ViacomCBS]]}}&lt;/ref&gt; In September 2019, Jordan confirmed that ''Creed III'' was officially in active development.&lt;ref&gt;{{Cite AV media |url=https://www.youtube.com/watch?v=ontvN-rEW38 |title=Michael B. Jordan Teases ''Creed III'' |date=2019-07-06 |publisher=[[ET Canada]] |access-date=2021-02-09 |archive-url=https://web.archive.org/web/20201107121144/https://www.youtube.com/watch?v=ontvN-rEW38 |archive-date=2020-11-07 |url-status=live |via=[[YouTube]]}}&lt;/ref&gt;
In February 2020, [[Zach Baylin]] was announced as screenwriter, with Jordan confirmed to reprise his role as [[Adonis Creed]].&lt;ref&gt;{{Cite web |last=Galuppo |first=Mia |date=February 25, 2020 |title=''Creed 3'' Taps 'King Richard' Writer (Exclusive) |url=https://www.hollywoodreporter.com/heat-vision/creed-3-taps-king-richard-writer-zach-baylin-1281094 |url-status=live |archive-url=https://web.archive.org/web/20200225205647/https://www.hollywoodreporter.com/heat-vision/creed-3-taps-king-richard-writer-zach-baylin-1281094 |archive-date=February 25, 2020 |access-date=February 25, 2020 |website=[[The Hollywood Reporter]]}}&lt;/ref&gt; In October 2020, it was reported that Jordan would reprise his role of Adonis Creed and make his [[directorial debut]] in ''Creed III''.&lt;ref&gt;{{Cite web |last=Fleming |first=Mike Jr. |date=October 24, 2020 |title=Don't Hold Your Breath Waiting To See 007 Film ''No Time To Die'' On Streamer As Cursory Talks Died Quickly |url=https://deadline.com/2020/10/james-bond-no-time-to-die-streamer-talks-died-quickly-1234602960/ |url-status=live |archive-url=https://web.archive.org/web/20201024201342/https://deadline.com/2020/10/james-bond-no-time-to-die-streamer-talks-died-quickly-1234602960/ |archive-date=October 24, 2020 |access-date=October 24, 2020 |website=[[Deadline Hollywood]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Gemmill |first=Allie |date=2020-10-24 |title=Will Michael B. Jordan Make His Directorial Debut With ''Creed 3''? |url=https://collider.com/creed-3-michael-b-jordan-director/ |url-status=live |archive-url=https://web.archive.org/web/20201025095139/https://collider.com/creed-3-michael-b-jordan-director/ |archive-date=October 25, 2020 |access-date=2020-10-24 |website=[[Collider (website)|Collider]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Ilyas |first=Xavier |date=October 24, 2020 |title=Michael B. Jordan Reportedly Set to Star &amp; Direct ''Creed 3'' |url=https://screenrant.com/creed-3-movie-michael-b-jordan-direct-rocky/ |url-status=live |archive-url=https://web.archive.org/web/20201129095717/https://screenrant.com/creed-3-movie-michael-b-jordan-direct-rocky/ |archive-date=November 29, 2020 |access-date=February 9, 2021 |website=[[Screen Rant]]}}&lt;/ref&gt; Producers had expressed interest in having Jordan serve as director, with [[Irwin Winkler]] stating that he had personally offered the position to the actor.&lt;ref&gt;{{Cite web |last=Ellis |first=Philip |date=September 6, 2019 |title=Michael B. Jordan Got an Offer to Direct ''Creed III'' If and When It Happens |url=https://www.menshealth.com/entertainment/a28938206/michael-b-jordan-might-direct-creed-3/ |url-status=live |archive-url=https://web.archive.org/web/20210126123617/https://www.menshealth.com/entertainment/a28938206/michael-b-jordan-might-direct-creed-3/ |archive-date=January 26, 2021 |access-date=February 9, 2021 |website=[[Men's Health]]}}&lt;/ref&gt; In April 2021, Stallone announced he was not cast in the film.&lt;ref&gt;{{Cite web |last=Bumbray |first=Chris |date=April 5, 2021 |title=''Creed 3'': Sylvester Stallone's Rocky sitting this round out? |url=https://www.joblo.com/movie-news/creed-3-sylvester-stallone-rocky |url-status=live |archive-url=https://web.archive.org/web/20210405194513/https://www.joblo.com/movie-news/creed-3-sylvester-stallone-rocky |archive-date=April 5, 2021 |access-date=April 5, 2021 |website=[[JoBlo.com]]}}&lt;/ref&gt; By June 2021, Jonathan Majors entered talks to portray Adonis's new adversary.&lt;ref&gt;{{Cite web |last=Vlessing |first=Etan |date=June 2, 2021 |title=Jonathan Majors in Talks to Join Michael B. Jordan in MGM's ''Creed III'' |url=https://www.hollywoodreporter.com/movies/movie-news/jonathan-majors-michael-b-jordan-mgms-creed-iii-1234961974/ |url-status=live |archive-url=https://web.archive.org/web/20210602180019/https://www.hollywoodreporter.com/movies/movie-news/jonathan-majors-michael-b-jordan-mgms-creed-iii-1234961974/ |archive-date=June 2, 2021 |access-date=August 20, 2021 |website=[[The Hollywood Reporter]]}}&lt;/ref&gt; In November 2021, it was officially confirmed that Majors was cast.&lt;ref&gt;{{Cite web |last=Jackson |first=Angelique |date=2021-11-04 |title=Jonathan Majors on ''The Harder They Fall'', ''Creed'' and His MCU Future as Kang the Conqueror |url=https://variety.com/2021/film/features/jonathan-majors-the-harder-they-fall-kang-creed-1235104429/ |url-status=live |archive-url=https://web.archive.org/web/20211104202334/https://variety.com/2021/film/features/jonathan-majors-the-harder-they-fall-kang-creed-1235104429/ |archive-date=November 4, 2021 |access-date=November 6, 2021 |website=[[Variety (website)|Variety]]}}&lt;/ref&gt; In April 2022, it was announced that [[Wood Harris]] and [[Florian Munteanu]] would reprise their roles from previous ''Creed'' films, and [[Selenis Leyva]], Thaddeus J. Mixson, Spence Moore II, and Mila Davis-Kent joined the cast.&lt;ref&gt;{{Cite web |last=Jackson |first=Angelique |date=April 7, 2022 |title=MGM's ''Creed III'' Casts Selenis Leyva, Thaddeus J. Mixson, Spence Moore II and Mila Davis-Kent (EXCLUSIVE) |url=https://variety.com/2022/film/news/creed-3-selenis-leyva-
```

thaddeus-j-mixson-spence-moore-ii-mila-davis-kent-1235227169/ |url-status=live |archive-url=https://web.archive.org/web/20220901174954/https://variety.com/2022/film/news/creed-3-selenis-leyva-thaddeus-j-mixson-spence-moore-ii-mila-davis-kent-1235227169/ |archive-date=September 1, 2022 |access-date=September 17, 2022 |website=[[Variety (magazine)|Variety]]}}&lt;/ref&gt; In September 2022, Mexican boxer [[Canelo Álvarez]] was cast in as a cameo.&lt;ref&gt;{{Cite web |last=Pattle |first=Alex |date=September 16, 2022 |title=Boxing champion Canelo Alvarez to star in ''Creed 3'', Michael B Jordan reveals |url=https://www.independent.co.uk/sport/boxing/canelo-creed-3-cast-michael-b-jordan-b2168999.html |url-status=live |archive-url=https://web.archive.org/web/20220917035031/https://www.independent.co.uk/sport/boxing/canelo-creed-3-cast-michael-b-jordan-b2168999.html |archive-date=September 17, 2022 |access-date=September 17, 2022 |website=[[The Independent]]}}&lt;/ref&gt;

3225 Stallone expressed distaste for the darker direction the filmmakers wanted to take the film in, stating: &quot;That's a regretful situation because I know what it could have been. It was taken in a direction that is quite different than I would've taken it. It's a different philosophy - Irwin Winkler's and Michael B. Jordan's. I wish them well, but I'm much more of a sentimentalist. I like my heroes getting beat up, but I just don't want them going into that dark space. I just feel people have enough darkness.&quot;&lt;ref&gt;{{Cite web |last=Hibberd |first=James |date=2022-11-07 |title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot; |url=https://www.hollywoodreporter.com/feature/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/ |access-date=2024-02-11 |website=The Hollywood Reporter |language=en-US}}&lt;/ref&gt; He eventually refused to see the film when it was released due to the involvement of producer [[Irwin Winkler]], whom Stallone has called a &quot;parasite&quot; for &quot;picking clean the bones&quot; of characters he (Stallone) created. Winkler purchased the rights for the Rocky character from Stallone in 1976.&lt;ref&gt;{{Cite web |date=2022-11-19 |title=Sylvester Stallone Explains Why He Will 'Never' Watch 'Creed III' |url=https://www.menshealth.com/entertainment/a42008229/sylvester-stallone-explains-why-he-will-never-watch-creed-iii/ |access-date=2024-02-11 |website=Men's Health |language=en-US}}&lt;/ref&gt; Jordan meanwhile wanted to explore Adonis' early life &amp; backstory.&lt;ref name=&quot;Mannix&quot;&gt;{{Cite magazine |last=Mannix |first=Chris |date=2023-03-03 |title=Michael B. Jordan Q&amp;A: Why Sylvester Stallone Isn't in 'Creed III' |url=https://www.si.com/boxing/2023/03/03/creed-iii-michael-b-jordan-director-rocky-franchise-sylvester-stallone |access-date=2024-02-11 |magazine=Sports Illustrated |language=en-us}}&lt;/ref&gt;

3226 | [[Deadly Prey]] || 1987 || [[Rumble in the Jungle|Rambo in the Bamboo]] - [[Sylvester Stallone|Stallone]] Klone to the Bone || for every death that our hero Mike Danton causes || 2020-05-22

3227 | '''''[[Guardians of the Galaxy Vol. 3]]''''' || [[James Gunn]] || [[Chris Pratt]], [[Zoe Saldaña]], [[Dave Bautista]], [[Vin Diesel]], [[Bradley Cooper]], [[Karen Gillan]], [[Pom Klementieff]], [[Elizabeth Debicki]], [[Sean Gunn]], [[Sylvester Stallone]], [[Will Poulter]], [[Chukwudi Iwuji]], [[Maria Bakalova]] || United States ||Superhero Comedy&lt;ref&gt;{{cite web|first=Andreas|last=Wiseman|title='Borat' Breakout Maria Bakalova Set For Key Role In Marvel's 'Guardians Of The Galaxy Vol. 3'|url=https://deadline.com/2022/06/guardians-galaxy-marvel-maria-bakalova-james-gunn-disney-borat-1235038826/|date=June 6, 2022|publisher=Deadline Hollywood|access-date=August 4, 2022}}&lt;/ref&gt;

3228 * [[Sylvester Stallone]] - actor&lt;ref&gt;{{cite news|title=Take a look inside Palm Beach property purchased by Sylvester Stallone |url=https://www.wptv.com/entertainment/entertainment-photo-gallery/take-a-look-inside-palm-beach-property-purchased-by-sylvester-stallone}}&lt;/ref&gt;

3229 | starring   = {{unbulleted list|[[Sylvester Stallone]]|[[Talia Shire]]|[[Burt Young]]|[[Burgess Meredith]]|[[Carl Weathers]]}}

3230 | narrator   = {{unbulleted list|Sylvester Stallone}}

3231 The film tells the story of [[Sylvester Stallone]] and the creation of the first Rocky film when he was an unknown actor. It is narrated by Sylvester Stallone as he takes the audience through rare and never-before-seen home movies, rehearsal footage and behind the scenes footage, giving insight into how the low budget fan-favorite film was made.&lt;ref name=Alex&gt;{{Cite news |url=https://www.usatoday.com/story/entertainment/movies/2020/06/08/narrator-sylvester-stallone-best-revelations-40-years-rocky-new-documentary/3150268001/ |title='40 Years of Rocky': Sylvester Stallone's best revelations in the new documentary |first=Bryan |last=Alexander |date=June 8, 2020 |work=[[USA Today]]}}&lt;/ref&gt;

3232 The film is narrated by Stallone with archive footage starring the Rocky cast, including [[Talia Shire]], [[Burt Young]], [[Carl Weathers]] and [[Burgess Meredith]].&lt;ref&gt;{{Cite web |url=https://www.cinemablend.com/news/2470107/sylvester-stallone-previews-40-years-of-rocky-documentary |title=Sylvester Stallone Previews 40 Years Of Rocky Documentary |last=Evans |first=Nick |date=April 10, 2019 |website=Cinema Blend}}&lt;/ref&gt;

3233 Talking about the making of the film, Johnson describes how the series of events conspired that meant he was hand picked by Stallone to create this documentary.&lt;ref&gt;{{Cite web |url=https://goingthedistance.libsyn.com/going-the-distance-derek-wayne-johnson-interview |title=Sylvester Stallone Fan Podcast Network: Going The Distance - Derek Wayne Johnson Interview |website=goingthedistance.libsyn.com}}&lt;/ref&gt;

3234 * Before getting into the prose, I have run across chatter online that [[Sylvester Stallone]] wanted to play Roger Grimes. This bit of casting information is mentioned in the following book ([https://www.google.com/books/edition/Massacred_By_Mother_Nature_Exploring_the/BtRNDwAAQBAJ link]), but I am uncertain if it would be considered a high-quality source.

3235 | founder   = [[Sylvester Stallone]]

3236 | key_people   = Sylvester Stallone (Co-Executive Chairmen)&lt;br /&gt;Braden Aftergood (Vice chairman)

3237 '''Balboa Productions''' is an American film and television [[production company]] founded and led by [[Sylvester Stallone]]. The studio is named after his character [[Rocky Balboa]] from the [[Rocky (franchise)|''Rocky'' franchise]].

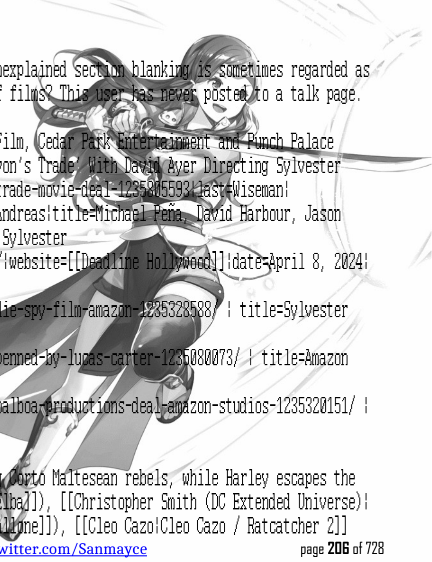
3238 On May 30, 2018, [[Sylvester Stallone]] formed Balboa Productions with a partnership with Starlight Culture Entertainment. Braden Aftergood was named as the executive of scripted development.&lt;ref name=&quot;JackJohnson&quot;&gt;{{Cite web|first=Mike Jr.|last=Fleming|title=Sly Stallone Launches Balboa Productions With Film On Trump-Pardoned Heavyweight Champ Jack Johnson|url=https://deadline.com/2018/05/sylvester-stallone-jack-johnson-donald-trump-balboa-productions-mgm-1202400002/|date=May 30, 2018|website=[[Deadline Hollywood]]|access-date=June 9, 2020}}&lt;/ref&gt; That same month, Balboa Productions announced one of their first projects with the superhero thriller ''[[Samaritan (film)|Samaritan]]''. Stallone starred in the titular role, with the movie being released on August 26, 2022, via streaming, through [[Prime Video|Amazon Prime Video]] after Amazon's purchase of MGM.&lt;ref name=&quot;fullslate&quot;&gt;{{Cite web|first=Amanda|last=N'Duka|title=Sylvester Stallone To Star In 'Samaritan' &amp; 'Hunter', Acquires FBI Memoir As His Balboa Productions Ramps Up Slate|url=https://deadline.com/2018/10/sylvester-stallone-samaritan-hunter-balboa-productions-ghost-book-deal-1202483446/|date=October 17, 2018|website=Deadline Hollywood|access-date=June 9, 2020}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|first=Anthony|last=D'Alessandro|title=Sylvester Stallone MGM Thriller 'Samaritan' Heads To Early December 2020|url=https://deadline.com/2019/11/sylvester-stallone-mgm-thriller-samaritan-opening-late-fall-2020-1202743588/|date=October 17, 2019|website=Deadline Hollywood|access-date=June 9, 2020}}&lt;/ref&gt; Balboa Productions was one of the studios involved with the development of ''[[Rambo: Last Blood]]'', released on September 20, 2019.&lt;ref&gt;{{Cite web|title=Rambo: Last Blood (2019 Movie) Teaser Trailer|url=https://www.youtube.com/watch?v=4vWg5yJuWfs|date=May 30, 2019|website=YouTube|access-date=June 9, 2020}}&lt;/ref&gt; In October 2019, Balboa Productions partnered with DAZN to develop sports featured documentaries. Their first collaboration was realized with, ''One Night: Joshua vs. Ruiz''.&lt;ref name=&quot;DAZN&quot;&gt;{{cite web|url=https://www.dazn.com/en-US/news/boxing/sylvester-stallone-teams-up-with-dazn-to-produce-one-night-joshua-vs-ruiz/1a1m7005xdprj17dd133m1d4k1|work=DAZN News|title=Sylvester Stallone Teams Up with DAZN to Produce 'One Night: Joshua vs. Ruiz'|author=Muehlhausen, Steven|date=October 24, 2019|accessdate=November 8, 2022}}&lt;/ref&gt; Balboa Productions entered a [[first-look deal]] with [[Amazon Studios]] in April 2023.&lt;ref&gt;{{cite web |last1=Andreeva |first1=Nellie |title=Sylvester Stallone &amp; His Balboa Productions Inks Deal With Amazon Studios |url=https://deadline.com/2023/04/sylvester-stallone-balboa-productions-deal-amazon-studios-1235320151/ |website=[[Deadline Hollywood]] |publisher=[[Penske Media Corporation]] |date=April 12, 2023}}&lt;/ref&gt;

3239 In May 2018, Balboa Productions announced to be developing a number of projects, including: a [[Jack Johnson (boxer)|Jack Johnson]] biopic,&lt;ref name=&quot;JackJohnson&quot; /&gt; film adaptations of [[James Byron Huggins]] novel ''[[Hunter (Huggins novel)|Hunter]]'', and Michael McGowan and Ralph Pezzullo's memoir ''Ghost''. The studio will also develop a [[United States Special Operations Command|Special Ops]] film written by retired Army Ranger Max Adams, alongside television series adaptations of [[Chuck Dixon]]'s ''Levon's Trade'', and Charles Sailor's ''Second Son''.&lt;ref name=&quot;fullslate&quot; /&gt; In May 2019, it was announced that Balboa Productions will co-produce the American remake of ''[[The Gangster, The Cop, The Devil]]'' with B&amp;C Group and CA Entertainment. [[Ma Dong-seok|Don Lee]] will produce the project, in addition to reprising his starring role in the remake.&lt;ref name=&quot;GCD&quot;&gt;{{Cite web|first=Leo|last=Hyo-won|title=Sylvester Stallone to Remake South Korean Thriller 'The Gangster, The Cop, The Devil'|url=https://www.hollywoodreporter.com/news/balboa-productions-remake-south-korean-thriller-gangster-cop-devil-1207585|date=May 5, 2019|website=[[The Hollywood Reporter]]|access-date=June 9, 2020}}&lt;/ref&gt;

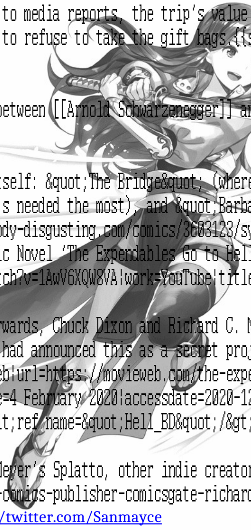
3240 In July 2019, it was announced that the studio will produce another slate of projects including: Corin Hardy's monster movie titled ''Arcane'', ''The Bellhop'' starring [[Iko Uwais]], and a television series adaptation of Stallone's [[Cult film|cult classic]] ''[[Nighthawks (1981 film)|Nighthawks]]''. The series will be released via [[Streaming media|streaming]] exclusively on [[Peacock (streaming service)|Peacock]]. Balboa Productions will also be involved in the upcoming [[History (American TV network)|History Channel Original]] series, ''[[Charles Becker|The Tenderloin]]'' with Stallone set to direct multiple episodes. As a production studio, they stated that they would like to be &quot;the Blumhouse of action films&quot;, referring to [[Blumhouse Productions]]' success in the [[Horror film|horror movie]] genre.&lt;ref name=&quot;moreprojects&quot;&gt;{{Cite web|first=Matt|last=Donnelly|title=Sylvester Stallone's Production Company Wants to Be the Blumhouse of Action Films|url=https://variety.com/2019/film/features/balboa-prods-sylvester-stallone-braden-aftergood-1203275756/|date=July 24, 2019|website=[[Variety (magazine)|Variety]]|access-date=June 9, 2020}}&lt;/ref&gt;&lt;ref name=&quot;Peacock&quot;&gt;{{Cite web|first=Sylvester|last=Stallone|title=Q&amp;A: Answering the Audience|url=https://www.instagram.com/tv/B_vmmY5pZvb/|date=May 3, 2020|website=[[Instagram]]|access-date=June 9, 2020}}&lt;/ref&gt;&lt;ref name=&quot;tenderloin&quot;&gt;{{Cite web|first=Dade|last=Hayes|title=History Developing Sylvester Stallone Period Cop Drama 'The Tenderloin' - Upfront|url=https://deadline.com/2019/03/history-developing-sylvester-stallone-period-cop-drama-the-tenderloin-upfronts-1202583817/|date=March 27, 2019|website=Deadline Hollywood|access-date=June 9, 2020}}&lt;/ref&gt;

3241 In June 2019, it was announced that Balboa Productions will produce a new television series titled, ''The International''. [[Dolph Lundgren]] will star in the main role, with Ken Sanzel serving as showrunner, and Stallone scheduled to direct the pilot episode. The series will be released on the [[CBS]] network.&lt;ref name=&quot;theinternational&quot;&gt;{{Cite web|first=Nellie|last=Andreeva|title=Dolph Lundgren-Sylvester Stallone Action Drama Series Heats Up TV Marketplace|url=https://deadline.com/2019/08/dolph-lundgren-sylvester-stallone-action-drama-series-interest-cbs-fox-nbc-netflix-apple-1202659482/|date=August 2, 2019|website=Deadline Hollywood|access-date=June 9, 2020}}&lt;/ref&gt;&lt;ref name=&quot;CBSInternational&quot;&gt;{{Cite web|first=Nellie|last=Andreeva|title=Dolph Lundgren-Sylvester Stallone Action Drama 'The International' Lands At CBS As Put Pilot|url=https://deadline.com/2019/10/dolph-lundgren-sylvester-stallone-action-drama-the-international-cbs-put-pilot-1202709813/|date=October 3, 2019|website=Deadline Hollywood|access-date=June 9, 2020}}&lt;/ref&gt; In February 2020, it was announced that the studio will produce Rowan Athale's film ''Little America''. Stallone was slated to appear in the starring role, while additional funding for the project is being acquired through AGC International.&lt;ref name=&quot;LittleAmerica&quot;&gt;{{Cite web|first=Andreas|last=Wiseman|title=Sylvester Stallone To Star In Dystopian Action Thriller 'Little America' With Michael Bay As EP; AGC To Launch Sales At EFM|url=https://deadline.com/2020/02/sylvester-stallone-star-action-movie-little-america-agc-efm-1202854971/|date=February 10, 2019|website=Deadline Hollywood|access-date=June 9, 2020}}&lt;/ref&gt;&lt;ref name=&quot;Deals&quot;&gt;{{Cite web|first=Andreas|last=Wiseman|title=Sylvester Stallone Sci Fi-Action Pic 'Little America' Pre-Sells To Raft Of Major Territories For AGC At EFM|url=https://deadline.com/2020/02/sylvester-stallone-sci-fi-action-pic-little-america-pre-sells-to-raft-of-major-territories-for-agc-at-efm-1202866699/|date=February 24, 2020|website=Deadline Hollywood|access-date=June 9, 2020}}&lt;/ref&gt; The studio is also set to produce Cassandra Brooksbank's directiorial debut ''My Masterpiece'', starring [[Maria Bakalova]].&lt;ref name=&quot;my masterpiece&quot;&gt;{{Cite web|first1=Brent|last1=Lang|first2=Matt|last2=Donnelly|title=&quot;Borat 2&quot; Breakout Maria Bakalova Starring in &quot;My Masterpiece&quot; for Sylvester Stallone's Balboa Productions|url=https://variety.com/2023/film/markets-festivals/borat-2-maria-bakalova-my-masterpiece-sylvester-stallone-balboa-productions-1235618793/|date=May 19, 2023|website=Variety|

```
access-date=May 20, 2023}}&lt;/ref&gt; In May 2023, a legacy sequel to ''[[Cliffhanger (film)|Cliffhanger]]'' was announced to be in the works with Stallone returning to star in addition to serving as a producer.&lt;ref name=&quot;Sequel_Deadline&quot;&gt;{{cite web|url=https://deadline.com/2023/05/sylvester-stallone-set-for-cliffhanger-reboot-from-director-ric-roman-waugh-1235351858/|work=Deadline|title=Sylvester Stallone Set For 'Cliffhanger' Reboot From Director Ric Roman Waugh|author=Grobar, Matt|date=May 1, 2023|accessdate=May 4, 2023}}&lt;/ref&gt;
3242 | ''[[Lost on a Mountain in Maine]]''&lt;ref&gt;{{cite web|last=Bechara|first=Diego Ramos|title=Sylvester Stallone-Produced Feature 'Lost on a Mountain In Maine' Slated for November Release by Blue Fox Entertainment (EXCLUSIVE)|url=https://variety.com/2024/film/news/lost-on-a-mountain-in-maine-blue-fox-entertainment-sylvester-stallone-1236085210/|website=[[Variety (magazine)|Variety]]|date=July 25, 2024|access-date=July 25, 2024}}&lt;/ref&gt;
3243 | Distribution by [[Paramount Pictures]]&lt;ref&gt;{{cite news |last1=Tartaglione |first1=Nancy |title=Paramount To Remake Korean Thriller 'The Gangster, The Cop, The Devil'; Don Lee Reprising Original Role &amp; Producing Alongside Sylvester Stallone's Balboa |url=https://deadline.com/2022/06/paramount-remake-the-gangster-the-cop-the-devil-don-lee-sylvester-stallone-balboa-1235053453/ |access-date=17 March 2024 |work=Deadline Hollywood |publisher=Penske Media Corporation |date=June 28, 2022}}&lt;/ref&gt;
3244 {{Sylvester Stallone}}
3245 | ''[[Alarum (film)|'''Alarum''']]'' || [[Polish brothers|Michael Polish]] || [[Sylvester Stallone]], [[Scott Eastwood]], [[Willa Fitzgerald]], [[Mike Colter]], [[Ísis Valverde]], [[D. W. Moffett]] || United States ||Action crime thriller&lt;ref&gt;{{cite web|first=Alex|last=Ritman|title=Sylvester Stallone Action Thriller 'Alarum' Acquired by Signature for U.K. (EXCLUSIVE)|url=https://variety.com/2024/film/global/sylvester-stallones-alarum-signature-uk-1236005945/|date=May 17, 2024|work=Variety|access-date=June 6, 2024}}&lt;/ref&gt;
3246 | ''[[Armored (upcoming film)|'''Armored''']]'' || Justin Routt || [[Sylvester Stallone]], [[Jason Patric]], [[Josh Wiggins]], [[Dash Mihok]] || United States ||Action thriller&lt;ref&gt;{{cite web|author1=Meg James|author2=Amy Kaufman|title=After scandal, movie producer Randall Emmett is flying under the radar with a new name|url=https://www.latimes.com/entertainment-arts/business/story/2024-04-26/randall-emmett-ives-sylvester-stallone|date=April 26, 2024|website=[[LA Times]]|access-date=June 6, 2024}}&lt;/ref&gt;
3247 | '''''Arcane''''' || [[Corin Hardy]] || || United States ||Monster&lt;ref&gt;{{cite web|first=Matt|last=Donnelly|title=Sylvester Stallone's Production Company Wants to Be the Blumhouse of Action Films|url=https://variety.com/2019/film/features/balboa-prods-sylvester-stallone-braden-aftergood-1203275756/|date=July 24, 2019|website=Variety|access-date=December 8, 2020}}&lt;/ref&gt;
3248 |  '''''[[Run Sweetheart Run]]''''' || [[Shana Feste]] || [[Ella Balinska]], [[Pilou Asbæk]], [[Clark Gregg]], [[Aml Ameen]], [[Dayo Okeniyi]], [[Betsy Brandt]], [[Shohreh Aghdashloo]] || United States ||Horror-thriller ||&lt;ref&gt;{{cite web|first=Anthony|last=D'Alessandro|title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan', 'Good Night Oppy' &amp; More|url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/|date=August 1, 2022|website=Deadline Hollywood|access-date=August 4, 2022}}&lt;/ref&gt;
3249 ''[[Animal Crackers (2017 film)|Animal Crackers]]'' was directed by Sava and Tony Bancroft.&lt;ref name=&quot;:1&quot;&gt;{{Cite web |last=Odman |first=Sydney |date=2017-08-31 |title='Animal Crackers': Meet the Voices Behind the Animated Characters |url=https://www.hollywoodreporter.com/lists/animal-crackers-cast-meet-famous-voice-actors-1016826/ |access-date=2022-11-28 |website=The Hollywood Reporter |language=en-US}}&lt;/ref&gt; ''Animal Crackers'' aired on [[Netflix]]&lt;ref name=&quot;:03&quot;/&gt; on July 24, 2020&lt;ref&gt;{{Cite web |title='Animal Crackers': Release date, plot, voice cast, trailer and all you need to know about the animated fantasy film on Netflix |url=https://meaww.com/animal-crackers-release-date-plot-voice-cast-trailer-netflix-film-animated-fantasy |access-date=2022-12-05 |website=meaww.com|date=8 July 2020 }}&lt;/ref&gt; starring [[John Krasinski]] as Owen Huntington, and [[Emily Blunt]] as Zoe Huntington, [[Ian McKellen]] as Horatio Huntington, as well as [[Danny DeVito]] as Chesterfield, and [[Sylvester Stallone]] as Bulletman.&lt;ref name=&quot;:1&quot; /&gt;
3250 |[[Sylvester Stallone]] as a former Champ Car star.
3251 &lt;li&gt;{{cite book |last1=Siegel |first1=Scott |authorlink1=Scott Siegel |last2=Siegel |first2=Barbara |date=1990 |title=The Encyclopedia Of Hollywood |url=https://archive.org/details/encyclopediaofho00sieg/ |location=New York |publisher=[[Facts on File]] |pages=[https://archive.org/details/encyclopediaofho00sieg/page/379/ 379]-[https://archive.org/details/encyclopediaofho00sieg/page/382/ 382] |isbn=978-1-4381-3008-8 |accessdate=2020-06-22 }}&lt;p&gt;The encyclopedia has an entry titled &quot;Sex Symbols: Female&quot;. It lists these sex symbols: Theda Bara, Clara Bow, Joan Crawford, Greta Garbo, Marlene Dietrich, Jean Harlow, Hedy Lamarr, Betty Grable, Veronica Lake, Rita Hayworth, Lana Turner, Ava Gardner, Jane Russell, Marilyn Monroe, Grace Kelly, Elizabeth Taylor, Carol Lynley, Carrol Baker, Ann-Margret, Jane Fonda, Sophia Loren, Virna Lisi, Gina Lollobrigida, Ursula Andress, Elke Sommer, and Bo Derek.&lt;/p&gt;&lt;p&gt;The encyclopedia has an entry titled &quot;Sex Symbols: Male&quot;. It lists these sex symbols: Tyrone Power, Montgomery Clift, Rudolph Valentino, Charles Boyer, Cary Grant, Rock Hudson, Sylvester Stallone, Francis X. Bushman, Rudolph Valentino, John Gilbert, Clark Gable, Victor Mature, Burt Lancaster, Marlon Brando, Paul Newman, Robert Redford, Warren Beatty, Kevin Costner, Tom Cruise, John Travolta, and Richard Gere.&lt;/p&gt;&lt;/li&gt;
3252     <redirect title="Sylvester Stallone" />
3253       <text bytes="179" sha1="2v3mnskf7cfof5khitxltgo26osymlc" xml:space="preserve">#REDIRECT [[Sylvester Stallone#Subsequent success: 1978-1999]]
3254 Filming began in October 2021 in New Jersey.&lt;ref&gt;{{cite web|url=https://www.newjerseystage.com/articles/2021/10/11/film-starring-anne-hathaway-anthony-hopkins-and-jeremy-strong-seeks-nj-kids-between-12-17-in-bayonnejersey-city-area/|title=Film Starring Anne Hathaway, Anthony Hopkins, and Jeremy Strong Seeks NJ Kids Between 12-17 In Bayonne/Jersey City Area|date=October 11, 2021|website=New Jersey Stage|access-date=October 15, 2021|archive-date=November 17, 2021|archive-url=https://web.archive.org/web/20211117195002/https://www.newjerseystage.com/articles/2021/10/11/film-starring-anne-hathaway-anthony-hopkins-and-jeremy-strong-seeks-nj-kids-between-12-17-in-bayonnejersey-city-area/|url-status=live}}&lt;/ref&gt; It was initially expected to start in early 2021.&lt;ref&gt;{{cite web |last1=Welk |first1=Brian |title=Focus Features Closes Deal for James Gray's ''Armageddon Time'' (Exclusive) |url=https://www.thewrap.com/focus-features-in-talks-to-finance-and-distribute-james-grays-armageddon-time-with-cate-blanchett/ |website=[[TheWrap]] |access-date=21 December 2020 |date=July 28, 2020 |archive-date=October 18, 2021 |archive-url=https://web.archive.org/web/20211018190500/https://www.thewrap.com/focus-features-in-talks-to-finance-and-distribute-james-grays-armageddon-time-with-cate-blanchett/ |url-status=live }}&lt;/ref&gt; In October, it was reported that [[Anthony Hopkins]] and [[Jeremy Strong (actor)|Jeremy Strong]] would also star alongside newcomers Banks Repeta, Jaylin Webb, and Ryan Sell, with Hopkins and Strong replacing De Niro and Isaac, respectively.&lt;ref&gt;{{cite web|url=https://deadline.com/2021/10/anthony-hopkins-jeremy-strong-anne-hathaway-james-grays-armageddon-time-focus-1234854179/|title=Anthony Hopkins and Jeremy Strong Join Anne Hathaway In James Gray's ''Armageddon Time'' For Focus|date=October 12, 2021|first=Justin|last=Kroll|website=[[Deadline Hollywood]]|access-date=October 12, 2021|archive-date=October 12, 2021|archive-url=https://web.archive.org/web/20211012195435/https://deadline.com/2021/10/anthony-hopkins-jeremy-strong-anne-hathaway-james-grays-armageddon-time-focus-1234854179/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.thewrap.com/anthony-hopkins-jeremy-strong-armageddon-time-james-gray/|title=Anthony Hopkins and Jeremy Strong Join James Gray's ''Armageddon Time'' With Anne Hathaway|date=October 12, 2021|first=Brian|last=Welk|website=[[TheWrap]]|access-date=October 12, 2021|archive-date=October 18, 2021|archive-url=https://web.archive.org/web/20211018190500/https://www.thewrap.com/anthony-hopkins-jeremy-strong-armageddon-time-james-gray/|url-status=live}}&lt;/ref&gt; Production wrapped in December 2021, and Andrew Polk and [[Tovah Feldshuh]] were confirmed to star.&lt;ref&gt;{{cite web|url=https://deadline.com/2021/12/armageddon-time-andrew-polk-joins-james-gray-focus-features-film-1234900762/|title=''Armageddon Time'': Andrew Polk Boards James Gray's Focus Features Coming-Of-Age Film|date=December 22, 2021|first=Matt|last=Grobar|website=[[Deadline Hollywood]]|access-date=December 22, 2021|archive-date=December 22, 2021|archive-url=https://web.archive.org/web/20211222183610/https://deadline.com/2021/12/armageddon-time-andrew-polk-joins-james-gray-focus-features-film-1234900762/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last1=Haun |first1=Harry |title=The Book of Tovah: From Jewish Cheerleader to Yentl to Dr. Ruth |url=https://observer.com/2021/12/the-book-of-tovah-from-jewish-cheerleader-to-yentl-to-dr-ruth/ |website=[[Observer Media]] |access-date=1 April 2022 |date=December 3, 2021 |archive-date=July 15, 2022 |archive-url=https://web.archive.org/web/20220715233811/https://observer.com/2021/12/the-book-of-tovah-from-jewish-cheerleader-to-yentl-to-dr-ruth/ |url-status=live }}&lt;/ref&gt; [[Domenick Lombardozzi]] was revealed as part of the cast in March 2022.&lt;ref&gt;{{cite web |last1=Andreeva |first1=Nellie |title=''Tulsa King'': Max Casella, Domenick Lombardozzi, Vincent Piazza &amp; Jay Will Join Sylvester Stallone In Paramount+ Series |url=https://deadline.com/2022/03/tulsa-king-max-casella-domenick-lombardozzi-vincent-piazza-jay-cast-sylvester-stallone-paramount-plus-1234985490/ |website=[[Deadline Hollywood]] |access-date=1 April 2022 |date=March 24, 2022 |archive-date=April 15, 2022 |archive-url=https://web.archive.org/web/20220415095241/https://deadline.com/2022/03/tulsa-king-max-casella-domenick-lombardozzi-vincent-piazza-jay-cast-sylvester-stallone-paramount-plus-1234985490/ |url-status=live }}&lt;/ref&gt; [[Jessica Chastain]] was later revealed to have replaced Blanchett in a cameo role.&lt;ref&gt;{{Cite web |last1=Donnelly |first1=Matt |last2=Vary |first2=Adam B. |date=2022-05-19 |title=Jessica Chastain Plays Surprise Role as Maryanne Trump, Donald Trump's Sister, in ''Armageddon Time'' |url=https://variety.com/2022/film/news/jessica-chastain-maryanne-trump-armageddon-time-cannes-1235272251/ |access-date=2022-05-19 |website=[[Variety (magazine)|Variety]] |archive-date=July 17, 2022 |archive-url=https://web.archive.org/web/20220717013943/https://variety.com/2022/film/news/jessica-chastain-maryanne-trump-armageddon-time-cannes-1235272251/ |url-status=live }}&lt;/ref&gt;
3255 | Loosely based on The Death Match, with [[Sylvester Stallone]] and [[Michael Caine]]; released in North America as ''Victory''.
3256 '''Page:''' {{pagelinks|Sylvester Stallone}} &lt;br /&gt;
3257 :::[https://en.wikipedia.org/w/index.php?title=User_talk:Chatterjee95&amp;diff=970473079&amp;oldid=970173764 I have left a note] for the editor. Unexplained section blanking is sometimes regarded as vandalism. Does he sincerely think that a good way to improve our articles on [[Sylvester Stallone]] and [[Woody Allen]] is to remove their list of films? This user has never posted to a talk page. [[User:EdJohnston|EdJohnston]] ([[User talk:EdJohnston|talk]]) 14:20, 31 July 2020 (UTC)
3258 | U.S. distribution only under the Metro-Goldwyn-Mayer label; co-production with [[Black Bear Pictures|Black Bear]], [[Balboa Productions]], BlockFilm, Cedar Park Entertainment and Punch Palace Productions; international distribution under [[Amazon Prime Video]]&lt;ref&gt;{{cite web|title=Amazon MGM Lands Jason Statham Action-Thriller 'Levon's Trade' With David Ayer Directing Sylvester Stallone Script; Wide U.S. Theatrical Release Lined Up|url=https://deadline.com/2024/01/jason-statham-david-ayer-sylvester-stallone-amazon-levons-trade-movie-deal-1235805593|last=Wiseman|first=Andreas|website=Deadline Hollywood|date=January 26, 2024|access-date=January 26, 2024}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last=Wiseman|first=Andreas|title=Michael Peña, David Harbour, Jason Flemyng &amp; Arianna Rivas Join Jason Statham In David Ayer's 'Levon's Trade'; Amazon MGM Sets Release Date For Black Bear Action Pic Scripted By Sylvester Stallone|url=https://deadline.com/2024/04/david-ayer-jason-statham-slyvester-stallone-movie-levons-trade-begins-amazon-mgm-release-date-1235878520/|website=[[Deadline Hollywood]]|date=April 8, 2024|access-date=April 8, 2024}}&lt;/ref&gt;
3259 |co-production with [[Balboa Productions]]&lt;ref&gt;{{cite web | url=https://deadline.com/2023/04/sylvester-stallone-to-star-in-never-too-old-to-die-spy-film-amazon-1235328588/ | title=Sylvester Stallone to Star in &amp; Produce Amazon Studios' Action Comedy 'Never Too Old to Die' Penned by Brian Otting | date=17 April 2023 }}&lt;/ref&gt;
3260 |co-production with [[Balboa Productions]]&lt;ref&gt;{{cite web | url=https://deadline.com/2022/07/amazon-studios-remaking-german-actioner-plan-b-penned-by-lucas-carter-1235080073/ | title=Amazon Taps Lucas Carter to Pen Remake of German Actioner 'Plan B'; Sylvester Stallone Among Producers | date=28 July 2022 }}&lt;/ref&gt;
3261 |co-production with [[Metro-Goldwyn-Mayer]] and [[Balboa Productions]]&lt;ref&gt;{{cite web | url=https://deadline.com/2023/04/sylvester-stallone-balboa-productions-deal-amazon-studios-1235320151/ |title=Sylvester Stallone &amp; His Balboa Productions Inks Deal with Amazon Studios | date=12 April 2023 }}&lt;/ref&gt;
3262 * Sylvester Stallone also auditioned for a part in the film.[7]
3263 Due to the first team being largely a decoy, a second team, both neither knew about, enters the country undetected and finds a recovered Flag among Corto Maltesean rebels, while Harley escapes the local government and reunites with Flag and the team, which includes [[Robert DuBois (DC Extended Universe)|Robert DuBois / Bloodsport]] ([[Idris Elba]]), [[Christopher Smith (DC Extended Universe)|Christopher &quot;Chris&quot; Smith / Peacemaker]] ([[John Cena]]), [[Nanaue (DC Extended Universe)|Nanaue / King Shark]] (voice of [[Sylvester Stallone]]), [[Cleo Cazo|Cleo Cazo / Ratcatcher 2]]
```



```
([[Daniela Melchior]]) and her pet rat Sebastian (voice of [[Dee Bradley Baker]]), and [[Abner Krill (DC Extended Universe)|Abner Krill / Polka-Dot Man]] ([[David Dastmalchian]]). The new squad captures Project Starfish's head geneticist [[Thinker (DC Comics)|Dr. Gaius Grieves / Thinker]] ([[Peter Capaldi]]) and blackmails him to entering them into Jötunheim in exchange of sparing his life, where they encounter the said project's asset, the [[starfish]]-like alien creature, [[Starro the Conqueror]].
3264 | [[Escape to Victory]] (Steff)|| [[John Huston]] || [[Sylvester Stallone]], [[Michael Caine]], [[Pelé]]
3265 | [[Rocky III]] (Elis)|| [[Sylvester Stallone]] || Sylvester Stallone, [[Talia Shire]], [[Carl Weathers]]
3266 | [[Rocky IV]] (Steff)|| [[Sylvester Stallone]]|| Sylvester Stallone, [[Talia Shire]], [[Burt Young]]
3267 Critic [[Lav Diaz]] noted ''Chinatown''{{'}}s intense violence, which he thought was influenced by the films of [[Arnold Schwarzenegger]] and [[Sylvester Stallone]], and he expressed mixed feelings over the varying quality of the film's action scenes. The film received four [[FAMAS Award]] nominations, winning [[FAMAS Award for Best Picture|Best Picture - Action]] and Best Sound (Rolly Ruta).
3268 [[Lav Diaz]], writing for the ''[[Manila Standard]]'', saw the film as another Filipino work influenced by the intense violence of films starring [[Sylvester Stallone]] or [[Arnold Schwarzenegger]], stating that &quot;Children too, [are] riddled with bullets.&quot; He gave mixed feelings for the action sequences, praising the chase scene involving a car and the action scenes at the start and the end of the film, but criticizing others for being basically filler. Diaz considers the love scene between Revilla's character and his lover to be the film's most memorable moment.&lt;ref name=&quot;MSDiaz&quot;&gt;{{cite news|last=Diaz|first=Lav|authorlink=Lav Diaz|title=Isa pang Rambo|trans-title=Another Rambo|url=https://news.google.com/newspapers?nid=8cBNEdFwSQkC&amp;dat=19881217&amp;printsec=frontpage&amp;hl=en|accessdate=August 3, 2020|work=[[Manila Standard]]|publisher=Manila Standard News, Inc.|date=December 17, 1988|page=15|language=Filipino|quote=Pati bata, tadtad sa bala.}}&lt;/ref&gt;
3269 |''[[Expend4bles]]'' || {{#invoke:dts|main|2022}} || {{#invoke:dts|main|September 22, 2023}}&lt;ref&gt;{{#invoke:cite web||url=https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video|title=Expendables 4: Sylvester Stallone Looks Thoroughly Badass In New Set Video|first=Erik|last=Swann|work=[[CinemaBlend]] |date=October 12, 2021|access-date=October 14, 2021|url-status=live|archive-date=October 14, 2021|archive-url=https://web.archive.org/web/20211014065910/https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video}}&lt;/ref&gt;&lt;ref&gt;{{#invoke:cite web||url=https://deadline.com/2022/10/expendables-4-sebastian-maniscalco-about-my-father-release-dates-1235137654/ | title=Lionsgate's 'Expendables 4', Sebastian Maniscalco's 'About My Father' &amp; Adele Lim Comedy Get 2023 Release Dates |first=Anthony |last=D'Alessandro |work=[[Deadline Hollywood]]|date=October 6, 2022 |access-date=October 7, 2022 |archive-date=October 6, 2022 |archive-url=https://web.archive.org/web/20221006224340/https://deadline.com/2022/10/expendables-4-sebastian-maniscalco-about-my-father-release-dates-1235137654/ |url-status=live }}&lt;/ref&gt;
3270 *''[[Samaritan (film)|Samaritan]]''&lt;ref&gt;{{#invoke:cite web||url=https://deadline.com/2020/03/sylvester-stallone-movie-the-samaritan-coronavirus-hiatus-1202883810/|title=Sylvester Stallone Superhero Thriller 'Samaritan' Going On Two-Week Hiatus|first1=Anthony|last1=D'Alessandro|date=March 15, 2020|access-date=March 15, 2020|archive-url=https://web.archive.org/web/20200316000107/https://deadline.com/2020/03/sylvester-stallone-movie-the-samaritan-coronavirus-hiatus-1202883810/|archive-date=March 16, 2020|url-status=live}}&lt;/ref&gt;
3271     <title>Category:Family of Sylvester Stallone</title>
3272 [[Category:Sylvester Stallone|Family]]
3273       <text bytes="50" sha1="5s8m44x5cu3xqc3e04s5lyhsqc9avay" xml:space="preserve">{{Category redirect|Family of Sylvester Stallone}}</text>
3274 *&lt;span id=&quot;White Eagle Enterprises&quot;&gt;{{no redirect|1 = White Eagle Enterprises }}&lt;/span&gt; → [[:Sylvester Stallone]] &lt;span&gt;&amp;nbsp;&lt;span class=&quot;plainlinks lx&quot;&gt;([[Talk:White Eagle Enterprises|talk]] &lt;b&gt;·&lt;/b&gt; [[Special:WhatLinksHere/White Eagle Enterprises|links]] &lt;b&gt;·&lt;/b&gt; [//en.wikipedia.org/w/index.php?title=White_Eagle_Enterprises&amp;action=history history] &lt;b&gt;·&lt;/b&gt; [https://iw.toolforge.org/pageviews?start=2020-09-18&amp;end=2020-10-17&amp;project=en.wikipedia.org&amp;pages=White_Eagle_Enterprises stats])&lt;/span&gt;&lt;/span&gt;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&lt;small class=&quot;plainlinks&quot;&gt;&lt;nowiki&gt;[&lt;/nowiki&gt;&amp;nbsp;Closure:&amp;nbsp;''{{#ifeq:{{FULLPAGENAME}}|Wikipedia:Redirects for discussion|(@subpage)|[{{fullurl:White Eagle Enterprises|action=edit&amp;summary={{Urlencode:[[{{FULLPAGENAME}}#White Eagle Enterprises]] closed as keep}}}} keep]/[{{fullurl:White Eagle Enterprises|action=edit&amp;summary={{Urlencode:[[{{FULLPAGENAME}}#White Eagle Enterprises]] closed as retarget}}}} retarget]&lt;span class=&quot;sysop-show&quot;&gt;/[{{fullurl:White Eagle Enterprises|action=delete&amp;wpReason={{Urlencode:[[{{FULLPAGENAME}}#White Eagle Enterprises]] closed as delete}}&amp;wpMovetalk=1}} delete]&lt;/span&gt;}}''&amp;nbsp;]&lt;/small&gt;&amp;nbsp;
3275 [[Lav Diaz]], writing for the ''[[Manila Standard]]'', gave ''Iyo ang Batas'' a negative review, criticizing the film overall as excessively cliché (&quot;gasgas&quot;) and citing its title, story, action, and dialogue as examples. Diaz stated that &quot;you would feel like you have already seen this movie many times or even heard it in a number of bloody radio programs.&quot;&lt;ref name=&quot;MSDiaz&quot;&gt;{{cite news|last=Diaz|first=Lav|authorlink=Lav Diaz|title=Bratatatatat!|url=https://news.google.com/newspapers?nid=8cBNEdFwSQkC&amp;dat=19880925&amp;printsec=frontpage&amp;hl=en|accessdate=November 2, 2020|work=[[Manila Standard]]|publisher=Manila Standard News, Inc.|date=September 25, 1988|page=26|language=Filipino|quote=Mararamdaman mong napanood mo na ang pelikulang ito nang makailang beses o kaya'y napakinggan na sa ilang madugong programa sa radyo.}}&lt;/ref&gt; However, he noted that what can be considered the film's strength is its intense action, in the vein of films starring [[Arnold Schwarzenegger]] and [[Sylvester Stallone]].&lt;ref name=&quot;MSDiaz&quot;/&gt;
3276 &quot;'''Winner Takes It All  '''&quot; is a 1987 rock song written by record producer [[Giorgio Moroder]] and [[Tom Whitlock|Thomas Whitlock]] and recorded by [[Sammy Hagar]]. Originally was included in the [[soundtrack]] of the [[Sylvester Stallone]] movie ''[[Over the Top (1987 film)|Over the Top]]'', being the first track and second single from the album, released through [[CBS Records International|CBS Records]]. The song peaked at No. 3 on the ''Billboard [[Mainstream Rock (chart)|Album Rock Tracks]]'' chart&lt;ref&gt;[https://www.billboard.com/artist/sammy-hagar/chart-history/rtt/ Billloard Mainstream Rock Chart]&lt;/ref&gt; and No. 54 on their ''[[Billboard Hot 100|Hot 100]]'' chart.&lt;ref&gt;[https://www.billboard.com/artist/sammy-hagar/chart-history/hsi/ Billboard Hot 100 Chart ]&lt;/ref&gt; It appears in Hagar's 2004 compilation album ''[[The Essential Red Collection]]''.
3277 The music video opens with Hagar running barefoot to the camera and playing guitar. As he sings the song, clips from the film play throughout. In the end, Hagar [[arm wrestle]]s with [[Sylvester Stallone]]'s character, Lincoln Hawk, and Hagar wins.
3278 The movie was mentioned in the April 2019 episode of the Red Letter Media's ''Best of the Worst'' series.&lt;ref&gt;{{cite web|url=https://www.youtube.com/watch?v=C1jbIPhWziI|title=Our DVD and Blu-ray Collection|publisher=[[YouTube]]. [[Alphabet Inc.]]|date=April 17, 2019|access-date=November 3, 2020}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.imdb.com/title/tt10929158/movieconnections/?tab=mc&amp;ref_=tt_trv_cnn|archive-url=https://archive.today/20201103123946/https://www.imdb.com/title/tt10929158/movieconnections/?tab=mc&amp;ref_=tt_trv_cnn|url-status=dead|archive-date=2020-11-03|title=Best of the Worst (TV Series) - Our DVD and Blu-ray Collection (2019) - Connections|publisher=[[IMDb]]. [[Amazon (company)|Amazon.com, Inc.]]|access-date=November 3, 2020}}&lt;/ref&gt; It may have inspired two action thriller films with similar plots: 1998's ''[[Firestorm (1998 film)|Firestorm]]'',&lt;ref name=&quot;cult&quot;/&gt;{{rp|at=59:55}} and ''[[Into the Blue (2005 film)|Into the Blue]]'' starring [[Paul Walker]] and [[Jessica Alba]] (2005).&lt;ref name=&quot;cult&quot;/&gt;{{rp|at=56:05}} Soon after its premiere, in November 1996, the ''[[Los Angeles Times]]'' labelled Matt McColm as &quot;Hollywood's next generation action star,&quot; following in the footsteps of [[Arnold Schwarzenegger]] and [[Sylvester Stallone]].&lt;ref&gt;{{cite web|url=https://news.google.com/newspapers?nid=1454&amp;dat=19961130&amp;id=FWhSAAAAIBAJ&amp;pg=6818,6314399|title=MOVIE TRENDS / What's happened to all the shoot-'em-up films? Who's the next Sly or Jean-Claude?|work=[[Los Angeles Times]]|date=November 30, 1996|first=Robert W.|last=Welkos|access-date=November 4, 2020}}&lt;/ref&gt;
3279 In 2019, Trump himself posted a photograph featuring his head attached to the body of Sylvester Stallone playing the part of [[Rocky Balboa]]. In 2022, Trump released a set of [[Non-fungible token|digital trading cards]] that depicted him in a variety of hyper-masculine roles, including him as a muscular [[superhero]], attracting ridicule from critics.&lt;ref&gt;{{Cite news |last=Taylor |first=Josh |date=2022-12-17 |title=Donald Trump's digital trading card collection sells out in less than a day |language=en-GB |work=The Guardian |url=https://www.theguardian.com/us-news/2022/dec/17/donald-trumps-digital-trading-card-collection-sells-out-in-less-than-a-day |access-date=2023-08-25 |issn=0261-3077}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Tran |first=Ken |title=Trump's &quot;major announcement&quot; of NFT collection draws ridicule but sells out |url=https://www.usatoday.com/story/news/politics/2022/12/16/donald-trump-nft-trading-card-collection/10908893002/ |access-date=2023-08-25 |website=USA TODAY |language=en-US}}&lt;/ref&gt;
3280 |Guests=[[Angela Lansbury]], [[Sylvester Stallone]], Willie Barcena
3281 |Guests=[[Kevin Sorbo]], [[Sylvester Stallone]], Woody Brooks
3282 In 2016, 26 Oscar-nominated celebrities were offered 10-day trips to Israel funded by the Israeli government, among them [[Leonardo DiCaprio]], [[Sylvester Stallone]], [[Mark Rylance]], [[Cate Blanchett]], [[Jennifer Lawrence]], [[Matt Damon]], and [[Kate Winslet]]. The Israeli Tourism Minister said that the celebrities were &quot;leading opinion-formers who we are interested in hosting&quot; who would get the chance to &quot;experience the country first-hand&quot;.{{sfn|Steinberg|2016}}{{sfn|''Reuters''|2016}} According to media reports, the trip's value was about $55,000 each but Israeli officials said that it was only about $15,000 to $18,000.{{sfn|Steinberg|2016}} Pro-Palestinian activists urged the celebrities to refuse to take the gift bags.{{sfn|''Reuters''|2016}} According to Catherine Rottenberg, not a single celebrity took advantage of the free trips.{{sfn|Rottenberg|2017}}
3283 * [[Sylvester Stallone]] (2015)
3284 Many sport rivalries exist as Wikipedia articles. But a rivalry between 2 actors was well covered by the media and was notable. The rivalry was between [[Arnold Schwarzenegger]] and [[Sylvester Stallone]]. I think it meets [[WP:N]], and request the [[WP:TH]] hosts to please present their views.
3285 |creators            = [[Sylvester Stallone]]
3286 There are three &quot;side quests&quot; in the story, where characters branch off the team in a subplot that is interwoven with the main story itself: &quot;The Bridge&quot; (where Barney Ross finds his still living friend Tool trapped in Hell), &quot;Christmas In Hell&quot; (where Lee Christmas is lured away from the team at a time where he's needed the most), and &quot;Barbarian's Holiday&quot; (where Gunner Jensen &quot;makes the most of a bad situation&quot;).&lt;ref name=&quot;Hell_BD&quot;&gt;{{cite web|url=https://bloody-disgusting.com/comics/3603123/sylvester-stallone-came-story-upcoming-graphic-novel-expendables-go-hell/|work=Bloody Disgusting|title=Sylvester Stallone Came Up With the Story for Upcoming Graphic Novel 'The Expendables Go to Hell'!|author=Squires, John|date=February 3, 2018|accessdate=October 7, 2021}}&lt;/ref&gt;&lt;ref name=&quot;Hell_CD&quot;&gt;{{cite web|url=https://www.youtube.com/watch?v=1AwV6XQW8VA|work=YouTube|title=The Expendables Go To Hell!|author=Dixon, Chuck|date=February 5, 2020|accessdate=October 7, 2021}}&lt;/ref&gt;
3287 [[Dynamite Entertainment]], which previously held the comic rights to ''The Expendables'', had let their license expire at an unknown date. Afterwards, Chuck Dixon and Richard C. Meyer made deals directly with [[Sylvester Stallone]] to get the rights to adapt the movie into a comic book. Meyer, known online as Comics MATTER w/Ya Boi Zack, had announced this as a secret project in July 2019 on his [[YouTube]] channel. In February 2020,  ''The Expendables Go to Hell'' was officially unveiled.&lt;ref name=&quot;movieweb&quot;&gt;{{cite web|url=https://movieweb.com/the-expendables-go-to-hell-comic-book-sylvester-stallone/|website=movieweb.com|title=The Expendables Go to Hell in New Graphic Novel Co-Developed by Sylvester Stallone|date=4 February 2020|accessdate=2020-12-31}}&lt;/ref&gt; The comic was crowdfunded on [[Indiegogo]], raising over $275,000 in total, and is set to release in 2021 under Meyer's Splatto Comics imprint.&lt;ref name=&quot;Hell_BD&quot;/&gt;&lt;ref name=&quot;Hell_CD&quot; /&gt;
3288 Early reports have also stated this will be published under a new indie label called Destination Comics, created to be a professional label for Meyer's Splatto, other indie creators and licensed comic books such as those featuring Stallone.&lt;ref name=&quot;bleedingcool&quot;&gt;{{cite web|url=https://bleedingcool.com/comics/destination-comics-publisher-comicsgate-richard-meyer-chuck-dixon-
```



sylvester-stallone-written-expendables/|title=Destination Comics - A New Publisher From Comicsgate's Richard Meyer and Chuck Dixon? With Sylvester Stallone-Written Expendables, And More|website=bleedingcool.com|date=6 February 2020|accessdate=2020-12-31}}&lt;/ref&gt;

3289 The concept and general story arc was conceptualized by [[Sylvester Stallone]], while [[Chuck Dixon]] and Richard C. Meyer co-wrote and developed the comic itself. The main storyline is drawn by [[Graham Nolan]], with the side quests drawn by Jason Johnson (&quot;The Bridge&quot;), Kelsey Shannon (&quot;Christmas in Hell&quot;) and [[Butch Guice]] (&quot;Barbarian's Holiday&quot;).The main cover was designed by Shannon, and with unique variant covers by [[Billy Tucci]], Meyer, Johnson, Renzo Rodriguez and [[Dave Dorman]].&lt;ref name=&quot;Hell_BD&quot; /&gt;&lt;ref name=&quot;Hell_CD&quot; /&gt;

3290 [[File:Sylvester Stallone &amp; Arnold Schwarzenegger (7588431980).jpg|thumb|Stallone (left) and Schwarzenegger (right) speaking at the 2012 [[San Diego Comic-Con]]|upright=1.35]]

3291 The rivalry between American actor [[Sylvester Stallone]] and Austrian-American actor [[Arnold Schwarzenegger]] went on for about twenty years, often involving incidents of [[One-upmanship|one-upsmanship]] and [[Deception|subterfuge]] from both parties.&lt;ref name=&quot;:3&quot;&gt;{{Cite web|date=2019-10-31|title=Sylvester Stallone and Arnold Schwarzenegger Are Feuding Because It's The 80s Again|url=https://screenrant.com/arnold-schwarzenegger-sylvester-stallone-feud/|access-date=2021-01-09 |website=ScreenRant}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|last1=Mitas|first1=Melissa|last2=Articles|first2=More |date=2020-11-19|title=Why Sylvester Stallone and Arnold Schwarzenegger Had a 'Violent Hatred' For Each Other|url=https://www.cheatsheet.com/entertainment/why-sylvester-stallone-and-arnold-schwarzenegger-had-a-violent-hatred-for-each-other.html/|access-date=2021-01-09|website=Showbiz Cheat Sheet}}&lt;/ref&gt;

3292 The two actors first met each other in 1977 when assigned to the same table at the [[34th Golden Globe Awards]], where ''[[Rocky]]'', written by and starring Stallone, had been nominated for six categories but came short in all but one of them, while Schwarzenegger was nominated for [[Golden Globe Award for New Star of the Year - Actor|New Star of the Year]], for ''[[Stay Hungry]]'', which he won.&lt;ref&gt;[https://www.goldenglobes.com/film/stay-hungry Stay Hungry|Golden Globes]&lt;/ref&gt; Stallone later spoke out in an interview with [[Variety (magazine)|''Variety'']], saying that Schwarzenegger laughed at him when ''Rocky'' lost in several categories.&lt;ref name=&quot;:0&quot;&gt;{{Cite web|date=2019-11-17|title=Arnold vs Sly: Schwarzenegger &amp; Stallone Have Been Feuding Since The 1980s|url=https://screenrant.com/arnold-schwarzenegger-sylvester-stallone-feud-explained/|access-date=2021-01-09|website=ScreenRant}}&lt;/ref&gt; Finally, when ''Rocky'' won [[Golden Globe Award for Best Motion Picture - Drama|Best Motion Picture - Drama]],&lt;ref&gt;[https://www.goldenglobes.com/film/rocky Rocky|Golden Globes]&lt;/ref&gt; Stallone threw a bowl of flowers at Schwarzenegger.&lt;ref&gt;{{Cite web|date=2019-08-13|title=That Time Sylvester Stallone Threw Flowers At Arnold Schwarzenegger At The Golden Globes|url=https://www.cinemablend.com/news/2478021/that-time-sylvester-stallone-threw-flowers-at-arnold-schwarzenegger-at-the-golden-globes|access-date=2021-01-09|website=CINEMABLEND}}&lt;/ref&gt;

3293 In ''[[Twins (1988 film)|Twins]],'' Schwarzenegger's character makes fun of a ''[[Rambo (franchise)|Rambo]]'' poster that he sees on the wall, implying with a hand gesture that he had bigger muscles than Stallone's depiction of Rambo. Stallone responded in the movie ''[[Tango &amp; Cash]]'', in which he beat up a man who looked like Schwarzenegger. Stallone later admitted that he imagined Schwarzenegger while shooting the scene.{{r|raymond20131018}} In ''[[Last Action Hero]]'', a ''[[Terminator 2: Judgment Day|Terminator 2]]'' poster was shown with Stallone's face, and Schwarzenegger's character compliments his performance in a tongue-in-cheek manner.&lt;ref&gt;{{Cite web |date=2020-07-07 |title=Last Action Hero: The Time Sylvester Stallone Was The Star of Terminator 2: Judgment Day |url=https://www.theterminatorfans.com/last-action-hero-the-time-sylvester-stallone-was-the-star-of-terminator-2-judgment-day/ |access-date=2022-04-14 |website=www.theterminatorfans.com}}&lt;/ref&gt; In the 1993 film ''[[Demolition Man (film)|Demolition Man]]'', John Spartan (Stallone) discovers that Schwarzenegger has a [[Presidential library|Presidential Library]] by the year 2032.&lt;ref name=&quot;:3&quot; /&gt;

3294 In an interview with [[The Tonight Show Starring Jimmy Fallon|Jimmy Fallon]], Stallone stated that Schwarzenegger once tricked him to do a terrible movie: the 1992 ''[[Stop! Or My Mom Will Shoot]]''.&lt;ref&gt;{{Cite web|title=Stallone Blames Schwarzenegger for His Most Regrettable Roles |url=https://www.hollywoodreporter.com/news/sylvester-stallone-reveals-how-arnold-725735|access-date=2021-01-10 |website=www.hollywoodreporter.com|date=15 August 2014 }}&lt;/ref&gt;&lt;ref&gt;{{Cite magazine|title=Watch Sylvester Stallone talk about old rivalry with Arnold Schwarzenegger |url=https://time.com/3116461/sylvester-stallone-arnold-schwarzenegger/|access-date=2021-01-10|magazine=Time}}&lt;/ref&gt; Schwarzenegger later admitted that it was true in an interview with [[Jimmy Kimmel Live!|Jimmy Kimmel]].&lt;ref name=&quot;:1&quot;&gt;{{Cite web|date=2019-11-03|title=Arnold Schwarzenegger Once Convinced Sylvester Stallone To Do A &quot;Piece Of Sh**&quot; Movie|url=https://www.animatedtimes.com/arnold-schwarzenegger-once-convinced-sylvester-stallone-to-do-a-piece-of-sh-movie/|access-date=2021-01-10|website=Animated Times}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|date=2019-10-29|title=Arnold Schwarzenegger Tricked Sylvester Stallone Into Making the 'Piece of S---' 'Stop! Or My Mom Will Shoot' (Video)|url=https://www.thewrap.com/arnold-schwarzenegger-tricked-sylvester-stallone-into-making-the-piece-of-s-stop-or-my-mom-will-shoot-video/|access-date=2021-01-10|website=TheWrap}}&lt;/ref&gt; Schwarzenegger said:{{blockquote|I read the script, and it was a piece of shit. Let's be honest. I say to myself, 'I'm not going to do this movie...' Then they went to Sly, and Sly called me (and asked), 'have they ever talked to you about doing this movie?' And I said, 'yes, I was thinking about doing it. This is a really brilliant idea, this movie.' When he heard that, because he was in competition, he said, 'Whatever it takes, I'll do the movie.' And of course the movie went major into the toilet.&lt;ref name=&quot;:1&quot; /&gt;}}

3295 The rivalry ended in the late 1990s when both actors' impact on the box office had reduced significantly.&lt;ref name=&quot;:2&quot;&gt;{{Cite news|title=Sylvester Stallone and I hated each other: Arnold Schwarzenegger|work=The Economic Times|url=https://economictimes.indiatimes.com/magazines/panache/sylvester-stallone-and-i-hated-each-other-arnold-schwarzenegger/articleshow/45239608.cms|access-date=2021-01-10}}&lt;/ref&gt; At the turn of the millennium, ''[[The Hollywood Reporter]]'' said that the pair were ideating a joint acting venture. Schwarzenegger also invited Stallone for multiple inaugurations during his time as the [[Arnold Schwarzenegger#Governor of California|Governor of California]]; as a gesture of goodwill, Stallone had donated $15,000 to Schwarzenegger's re-election campaign in 2005. The pair also starred in three movies together: ''[[The Expendables 2]], [[The Expendables 3]]'', and ''[[Escape Plan (film)|Escape Plan]].'' Additionally, Schwarzenegger also had an uncredited cameo appearance in ''[[The Expendables (2010 film)|The Expendables]].''&lt;ref name=&quot;:2&quot; /&gt;&lt;ref&gt;{{Cite news |last=Genzlinger |first=Neil |date=2013-10-17|title=Behind Bars, Where Anything Goes |work=The New York Times |url=https://www.nytimes.com/2013/10/18/movies/schwarzenegger-and-stallone-star-in-escape-plan.html|access-date=2021-01-10|issn=0362-4331}}&lt;/ref&gt;

3296 [[Category:Sylvester Stallone]]

3297 | [[Sylvester Stallone]]&lt;ref name=&quot;StalloneGotG3&quot; /&gt;

3298 &lt;ref name=&quot;StalloneGotG3&quot;&gt;{{Cite web |last=DeVore |first=Britta |date=November 12, 2021 |title=Sylvester Stallone Confirms His Return to the MCU in 'Guardians of the Galaxy Vol. 3' |url=https://collider.com/guardians-of-the-galaxy-3-sylvester-stallone/ |url-status=live |archive-url=https://web.archive.org/web/20211113035650/https://collider.com/guardians-of-the-galaxy-3-sylvester-stallone/ |archive-date=November 13, 2021 |access-date=November 12, 2021 |website=[[Collider (website)|Collider]]}}&lt;/ref&gt;

3299 As casting started to take place, and Alexander Salkind wanted a famous actor in the role of [[Superman]]. [[Dustin Hoffman]] was once again highly considered though the idea was dropped. [[Robert Redford]] was offered the part, but he felt he was too famous and was not right for the role, as did [[Burt Reynolds]]. After the success of ''[[Rocky]]'', [[Sylvester Stallone]] lobbied hard for the role but was ignored. [[Paul Newman]] was offered all three roles of Superman, Jor-El and Lex Luthor&lt;ref&gt;{{cite web |url=http://www.warpedfactor.com/2014/08/10-things-you-may-not-know-about.html|title=10 Things you Might Not Know About SUPERMAN: THE MOVIE|last= |first= |date=February 4, 2020|website=Warped Factor|publisher= |access-date= |quote=}}&lt;/ref&gt; to his choosing though he did not accept any of them, for a salary of $4 million. [[Nick Nolte]] and [[Rex Harrison]] would both turn it down. It was decided to focus on casting [[Jor-El]] and [[Lex Luthor]] and save Superman for later. Hoffman was then offered the role of Luthor, and although he was interested, he turned it down. [[Marlon Brando]] would eventually be cast as Jor-El, causing [[Gene Hackman]] to be cast as Luthor as he wanted desperately to work with Brando. Impressed by the producers' ability to cast famous actors in respective roles, [[Warner Brothers|Warner Bros.]] decided to distribute the film internationally, rather than simply [[Domestic market|domestically]]. The filmmakers made it a priority to shoot all of Brando's and Hackman's footage &quot;because they would be committed to other films immediately.&quot;

3300 '''''Face to Face with &quot;Weird Al&quot; Yankovic''''' was an American [[YouTube]] comedy [[web series]] hosted/written by comedian [[&quot;Weird Al&quot; Yankovic]].&lt;ref&gt;{{Cite web |title=&quot;Weird Al&quot; Yankovic |url=https://www.weirdal.com/news/face-to-face/}}&lt;/ref&gt; It ran on the YouTube channel Nerdist, from early April-mid August 2012. It was presented as a talk show. At the start of each episode, Yankovic would introduce the celebrity he interviewed that day. Usually, there would be two celebrity interviews per episode. The interviews featured footage from already-existing interviews with movie stars, with Yankovic edited into them, asking bizarre questions to comical answers from the celebrity. The celebrities he interviewed include [[Denzel Washington]], [[Megan Fox]], [[Uma Thurman]], [[Nicolas Cage]], [[Morgan Freeman]], [[Jesse Eisenberg]] (mistaken as [[Michael Cera]]), [[Sylvester Stallone]], [[Antonio Banderas]], [[Patrick Stewart]], [[Harrison Ford]], [[Rose McGowan]], [[Jeff Bridges]], [[Robert Pattinson]], [[Pierce Brosnan]], [[Julianne Moore]], [[Josh Brolin]], and [[William H. Macy]].

3301 Actors as Vin Diesel, Sylvester Stallone, Robin Williams, Chris Rock has difficult accent. Arnold has easy accent.

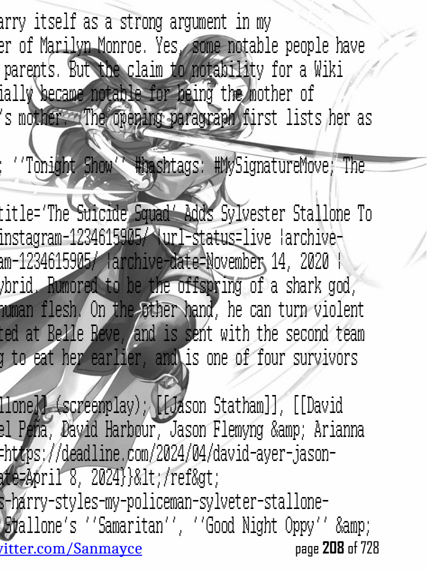
3302 :'''Comment''' - The assertion that &quot;a nomination to delete based on the fact that she was a relative of somebody who was more known doesn't carry itself as a strong argument in my opinion&quot;. My biggest issue with that statement is that Gladys Pearl Baker literally has no other claim to notability other than being the mother of Marilyn Monroe. Yes, some notable people have notable parents. Some notable people have parents that become notable after their children have become notable for something other than being their parents. But the claim to notability for a Wiki entry shouldn't be merely for being the parent of someone notable. You mentioned [[Jackie Stallone]] elsewhere. While Jackie Stallone may have initially became notable for being the mother of Sylvester Stallone, she became notable in her own right for other endeavors. Gladys Pearl Baker has only ever been notable for being Marilyn Monroe's mother. The opening paragraph first lists her as a film editor. She wasn't a notable film editor. [[User:ExRat|ExRat]] ([[User talk:ExRat|talk]]) 19:40, 18 January 2021 (UTC)

3303 |ShortSummary=Jimmy portrays Joe Biden in audio footage; [[Sylvester Stallone]]/Bob Dylan/Bruce Springsteen impromtu interview (Jimmy portrays all); ''Tonight Show'' #hashtags: #MySignatureMove; The Yahoo! Answers Lounge Singers (Jimmy Fallon &amp; Cristin Milioti); Rod Wave performed &quot;[[Tombstone (Rod Wave song)|Tombstone]]&quot;

3304 '''Nanaue''' (voiced by [[Sylvester Stallone]]),&lt;ref name=&quot;Stallone&quot;&gt;{{Cite web |last=Haring |first=Bruce |date=November 14, 2020 |title='The Suicide Squad' Adds Sylvester Stallone To Its Lineup, Confirmed By James Gunn On Instagram |url=https://deadline.com/2020/11/the-suicide-squad-adds-sylvester-stallone-james-gunn-confims-on-instagram-1234615905/ |url-status=live |archive-url=https://web.archive.org/web/20201114232924/https://deadline.com/2020/11/the-suicide-squad-adds-sylvester-stallone-james-gunn-confims-on-instagram-1234615905/ |archive-date=November 14, 2020 |access-date=November 14, 2020 |website=[[Deadline Hollywood]]}}&lt;/ref&gt; more commonly known as '''King Shark''', is a [[demigod]] shark-human hybrid. Rumored to be the offspring of a shark god, Nanaue is a kind-hearted but childish individual who is misunderstood due to his appearance, monosyllabic speech, ravenous appetite, and taste for human flesh. On the other hand, he can turn violent and aggressive to his opponents, often consuming them or tearing them to pieces. He had presumably turned to a life of crime before being incarcerated at Belle Reve, and is sent with the second team with Cleo, her pet rat Sebastian, Bloodsport, Peacemaker, and Polka-Dot Man to Corto Maltese. Nanaue forms a close bond with Cleo despite attempting to eat her earlier, and is one of four survivors on the team along with Cleo, Sebastian, Harley Quinn, and Bloodsport.

3305 | ''[[Levon's Trade]]'' || [[Metro-Goldwyn-Mayer]] / [[Black Bear Pictures]] / [[Balboa Productions]] || [[David Ayer]] (director); [[Sylvester Stallone]] (screenplay); [[Jason Statham]], [[David Harbour]], [[Michael Peña]], [[Jason Flemyng]] || style=&quot;text-align:center&quot; |&lt;ref&gt;{{cite web|last=Wiseman|first=Andreas|title=Michael Peña, David Harbour, Jason Flemyng &amp; Arianna Rivas Join Jason Statham In David Ayer's 'Levon's Trade'; Amazon MGM Sets Release Date For Black Bear Action Pic Scripted By Sylvester Stallone|url=https://deadline.com/2024/04/david-ayer-jason-statham-slyvester-stallone-movie-levons-trade-begins-amazon-mgm-release-date-1235878520/|website=[[Deadline Hollywood]]|date=April 8, 2024|access-date=April 8, 2024}}&lt;/ref&gt;

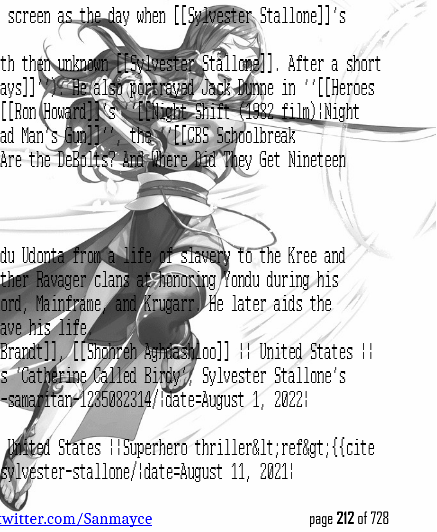
3306 The film premiered on [[Amazon Prime Video]] on November 18, 2022.&lt;ref&gt;{{cite web|url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/|title=Prime Video Dates Fall Movie Slate: Harry Styles' ''My Policeman'', Lena Dunham's ''Catherine Called Birdy'', Sylvester Stallone's ''Samaritan'', ''Good Night Oppy'' &amp;



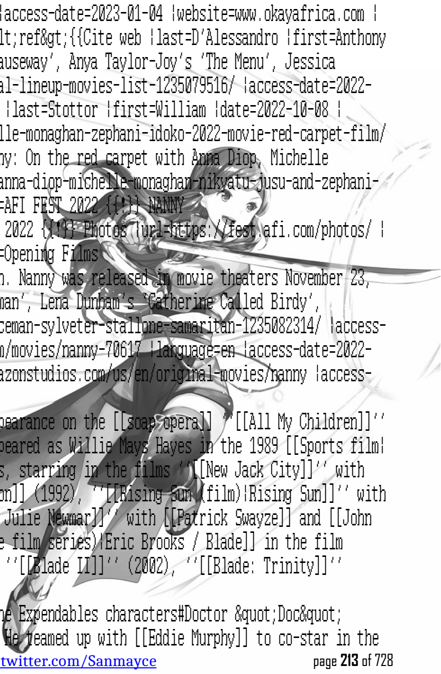
```
More|date=August 1, 2022|first=Anthony|last=D'Alessandro|website=[[Deadline Hollywood]]|access-date=August 3, 2022}}&lt;/ref&gt;
3307 [[Stakar Ogord (Marvel Cinematic Universe)|Stakar Ogord]] (portrayed by [[Sylvester Stallone]]) led his own team of Ravagers. Other members included [[Yondu Udonta (Marvel Cinematic Universe)|Yondu Udonta]], [[Aleta Ogord (Marvel Cinematic Universe)|Aleta Ogord]], [[Charlie-27 (Marvel Cinematic Universe)|Charlie-27]], [[Martinex (Marvel Cinematic Universe)|Martinex]], [[Mainframe (Marvel Cinematic Universe)|Mainframe]], and [[Krugarr (Marvel Cinematic Universe)|Krugarr]] (portrayed by [[Michael Rooker]], [[Michelle Yeoh]], [[Ving Rhames]], [[Michael Rosenbaum]], [[Miley Cyrus]] (later [[Tara Strong]]) respectively, with Krugarr being portrayed through CGI). The team eventually disbanded, but was reunited following the death of their former teammate Yondu.
3308 ** [[Sylvester Stallone]]&amp;nbsp;- ''[[Creed (film)|Creed]]''
3309 ''Nanny'' had its world premiere at the [[2022 Sundance Film Festival|Sundance Film Festival]] on January 22, 2022.&lt;ref&gt;{{Cite web|last1=D'Alessandro|first1=Anthony|last2=Patten|first2=Dominic|date=2021-12-09|title=Sundance 2022: Hybrid Festival Sees Princess Diana, Michael Kenneth Williams, Dakota Johnson, Bill Cosby, NYC Rock'n'Roll &amp; Regina Hall Fill Lineup|url=https://deadline.com/2021/12/sundance-film-festival-2022-lineup-films-competition-midnight-spotlight-kids-next-full-list-1234886826/|access-date=2021-12-09|website=Deadline Hollywood}}&lt;/ref&gt; In March 2022, [[Amazon Studios]] and [[Blumhouse Productions]] acquired distribution rights to the film in a deal around $7 million, winning them in a competitive situation that also included [[Sony Pictures Classics]] and [[Neon (company)|Neon]]. [[Jason Blum]], who joined the film as an executive producer post-acquisition, remarked: &quot;We're proud to have writer/director [[Nikyatu Jusu]]'s ''Nanny'' as part of our slate for Amazon. It's a gem of a horror film that combines impressive filmmaking and powerful storytelling, and is worthy of the Grand Jury Prize it was awarded at Sundance.&quot; The studios plan to release the film both theatrically and on [[Amazon Prime Video|Prime Video]].&lt;ref&gt;{{cite web|url=https://deadline.com/2022/03/blumhouse-horror-nanny-sundance-amazon-prime-1234980018/|title=Prime Video &amp; Blumhouse Win World Rights To Sundance-Winning Horror 'Nanny' In $7M Deal|website=Deadline Hollywood|first=Andreas|last=Wiseman|date=March 16, 2022|access-date=March 16, 2022}}&lt;/ref&gt; A special presentation of the film was screened at the [[2022 Toronto International Film Festival|Toronto International Film Festival]] in September 2022, followed by one at the 2022 [[AFI Fest]] on November 3, 2022.&lt;ref&gt;{{cite web|url=https://tiff.net/events/nanny |title=Nanny |publisher=[[Toronto International Film Festival]]}}&lt;/ref&gt; ''Nanny'' was given a [[limited theatrical release]] on November 23, 2022, by Amazon Studios, prior to streaming on [[Prime Video]] starting December 16, 2022.&lt;ref&gt;{{cite press release|url=https://press.amazonstudios.com/us/en/original-movies/nanny|title=Nanny|publisher=[[Amazon Studios]]|access-date=July 31, 2022}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/|title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan' &amp; More|website=Deadline Hollywood|first=Anthony|last=D'Alessandro|date=August 1, 2022|access-date=August 1, 2022}}&lt;/ref&gt;
3310 |&lt;ref&gt;{{cite web|first=Matt|last=Donnelly|title=Sylvester Stallone's Production Company Wants to Be the Blumhouse of Action Films|url=https://variety.com/2019/film/features/balboa-prods-sylvester-stallone-braden-aftergood-1203275756/|date=July 24, 2019|website=Variety|access-date=June 14, 2021}}&lt;/ref&gt;
3311 :While you're waiting, {{u|Factelf4}}, you might add references for {{DarkRed|Mariani lives in Tuckahoe, New York with his wife. He has two sons and three granddaughters. / John Mariani began his career writing for New York Magazine in 1973, covering arts and entertainment stories. He wrote for numerous other publications including The New York Times, Financial Times, and Chicago Tribune. / His cover story for New York magazine focused on changing focus on the much-used theme of &quot;Women in Jeopardy&quot;. He also wrote profiles and did interviews with film directors like Frank Capra, Stanley Donen and Richard Brooks and actors like Henry Fonda, Ann Miller, Sylvester Stallone, Jack Nicholson, Audrey Hepburn, Sam Waterston, Debbie Reynolds and Jon Voight.}} -- [[User:Hoary|Hoary]] ([[User talk:Hoary|talk]]) 13:02, 18 May 2021 (UTC)
3312 * [[Sylvester Stallone]]
3313 &lt;!-- STOP! Per the talk page and the billing block, this is the actual title, not a stylization --&gt;'''''Expend4bles'''''&lt;!-- STOP! Per the talk page and the billing block, this is the actual title, not a stylization --&gt; (also known as '''''The Expendables 4''''') is a 2023 American [[action comedy film]]&lt;ref&gt;{{cite news|title='Expend4bles': New cast members, more explosions and a nonsensical plot|url=https://www.washingtonpost.com/style/2023/09/21/expendables-4-movie-review/|newspaper=[[The Washington Post]]|date=21 September 2021|accessdate=29 January 2024|first=Mark|last=Jenkins|archive-date=29 January 2024|archive-url=https://web.archive.org/web/20240129105208/https://www.washingtonpost.com/style/2023/09/21/expendables-4-movie-review/}}&lt;/ref&gt; and the fourth installment in [[The Expendables (franchise)|''The Expendables'' franchise]], following ''[[The Expendables 3]]'' (2014). The film stars an [[ensemble cast]] of largely action film actors consisting of [[Jason Statham]], [[50 Cent|Curtis &quot;50 Cent&quot; Jackson]], [[Megan Fox]], [[Dolph Lundgren]], [[Tony Jaa]], [[Iko Uwais]], [[Randy Couture]], [[Jacob Scipio]], [[Levy Tran]], [[Andy García]], and [[Sylvester Stallone]]. It is directed by [[Scott Waugh]] from a screenplay by [[Kurt Wimmer]], Tad Daggerhart and [[Max D. Adams]], based on a story by [[Spenser Cohen]], Wimmer and Daggerhart.
3314 * [[Sylvester Stallone]] as Barney Ross: The leader of the Expendables. Stallone confirmed that the film would be his final appearance as Barney Ross, with Statham expecting to take over the series after his departure.&lt;ref name=&quot;Devore&quot;&gt;{{Cite web|last=Devore|first=Britta|date=October 16, 2021|title=Sylvester Stallone Finishes Filming 'The Expendables 4,' Says Goodbye to the Series|url=https://collider.com/sylvester-stallone-wraps-filming-expendables-4/|url-status=live|access-date=October 19, 2021|website=[[Collider (website)|Collider]]|archive-date=November 11, 2021|archive-url=https://web.archive.org/web/20211111020416/https://collider.com/sylvester-stallone-wraps-filming-expendables-4/}}&lt;/ref&gt; Due to this, his role in the film was limited.&lt;ref name=&quot;HollywoodReporter&quot;&gt;{{Cite web|last=Hibberd|first=James|date=November 7, 2022|title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot;|url=https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/|url-status=live|access-date=November 8, 2022|website=[[The Hollywood Reporter]]|archive-date=November 7, 2022|archive-url=https://web.archive.org/web/20221107235253/https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/}}&lt;/ref&gt;
3315 Additionally, Sheila Shah was cast as Mandy / Adele, alongside [[Eddie Hall]] as a bouncer.&lt;ref name=&quot;THR&quot; /&gt;&lt;ref name=&quot;OctoberCast&quot;&gt;{{Cite web|last=Bureau|first=Athens|date=October 4, 2021|title=Sylvester Stallone's The Expendables 4 to begin filming in Greece|url=https://greekcitytimes.com/2021/10/04/stallone-the-expendables-4/|url-status=live|access-date=October 19, 2021|website=Greek City Times|archive-date=October 4, 2021|archive-url=https://web.archive.org/web/20211004110333/https://greekcitytimes.com/2021/10/04/stallone-the-expendables-4/}}&lt;/ref&gt;
3316 After ''[[The Expendables 3]]'' (2014) underperformed at the box office, the future of the series was left in doubt&lt;ref name=&quot;Hogan&quot;&gt;{{cite web |url=https://www.cinemablend.com/new/Hulk-Hogan-Expendables-4-Here-What-He-Said-71640.html |work=[[CinemaBlend]] |title=Hulk Hogan For The Expendables 4? Here's What He Said |last=O'Connell |first=Sean |date=June 15, 2015 |access-date=July 1, 2022 |archive-date=July 1, 2022 |archive-url=https://web.archive.org/web/20220701133503/https://www.cinemablend.com/new/Hulk-Hogan-Expendables-4-Here-What-He-Said-71640.html |url-status=live }}&lt;/ref&gt; and production stalled over several years. By November 2014, it was announced that the project is being developed with intent to retain its R-rating like the first two films.&lt;ref&gt;{{cite web|url=https://screenrant.com/the-expendables-4-r-rating/|title='The Expendables 4' Will Be Rated R|work=Screen Rant|author=Keyes, Rob|date=November 23, 2014|access-date=April 20, 2021|archive-date=November 20, 2021|archive-url=https://web.archive.org/web/20211120022135/https://screenrant.com/the-expendables-4-r-rating/|url-status=live}}&lt;/ref&gt; In December 2016, [[Sylvester Stallone]] announced that the fourth installment will be the final film in the series, while a scheduled tentative release date was set for 2018.&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/news/splendid-takes-expendables-4-german-speaking-europe-957840|title=Splendid Takes 'The Expendables 4' for German-Speaking Europe|work=The Hollywood Reporter|author=Roxborough, Scott|date=December 20, 2016|access-date=April 20, 2021|archive-date=April 22, 2021|archive-url=https://web.archive.org/web/20210422183112/https://www.hollywoodreporter.com/news/splendid-takes-expendables-4-german-speaking-europe-957840|url-status=live}}&lt;/ref&gt; By March 2017, Stallone had left the project and the franchise, due to creative differences over the script and direction for continuing the franchise.&lt;ref&gt;{{cite web|url=https://deadline.com/2017/03/sylvester-stallone-departs-the-expendables-franchise-1202057171/|title=Sylvester Stallone Departs 'The Expendables' Franchise|work=Deadline|date=March 31, 2017|access-date=April 20, 2021|archive-date=December 18, 2021|archive-url=https://web.archive.org/web/20211218135913/https://deadline.com/2017/03/sylvester-stallone-departs-the-expendables-franchise-1202057171/|url-status=live}}&lt;/ref&gt; In January 2018, after vocal support from other cast members (including [[Arnold Schwarzenegger]]), Stallone announced his return to the series with a post to his social media platforms; confirming new developments on the fourth movie.&lt;ref&gt;{{cite web|url=https://screenrant.com/sylvester-stallone-expendables-4-next/|title=Sylvester Stallone Hinting at Expendables 4 Next?|date=January 5, 2018|work=Screen Rant|access-date=April 20, 2021|archive-date=January 6, 2018|archive-url=https://web.archive.org/web/20180106040241/https://screenrant.com/sylvester-stallone-expendables-4-next/|url-status=live}}&lt;/ref&gt;
3317 By August 2020, Vértice Cine announced their involvement as a production studio on the movie, alongside [[Lionsgate]] and [[Millennium Films]]. They also revealed that [[Patrick Hughes (director)|Patrick Hughes]] will return to the series as director.&lt;ref name=&quot;Vértice announcement&quot;&gt;{{cite web|url=https://www.twitter.com/vertice360_cine/status/1290966907013738498|title=Vertice Cine (@vertice360_cine) on Twitter: NOTICIÓN Os presentamos los próximos estrenos de Vértice!! ¿Cuál te apetece más?|date=August 5, 2020|publisher=Vértice Cine - official|access-date=April 20, 2021|archive-date=August 5, 2020|archive-url=https://web.archive.org/web/20200805110505/https://twitter.com/vertice360_cine/status/1290966907013738498|url-status=live}}&lt;/ref&gt; In November 2020, the president of Millennium Media, Jeffrey Greenstein, stated that the studio is continuing to work on ''The Expendables 4'' after various delays within the industry worldwide due to the [[COVID-19 pandemic]].&lt;ref&gt;{{cite web|url=https://variety.com/2020/film/news/gerard-butler-fallen-afm-1234818543/|work=[[Variety (magazine)|Variety]]|title=Millennium's Jeffrey Greenstein Unveils a Fourth 'Has Fallen' as AFM Launches|author=McNary, Dave|date=November 9, 2020|access-date=April 20, 2021|archive-date=November 9, 2020|archive-url=https://web.archive.org/web/20201109174723/https://variety.com/2020/film/news/gerard-butler-fallen-afm-1234818543/|url-status=live}}&lt;/ref&gt; In August 2021, ''[[The Hollywood Reporter]]'' reported that [[Scott Waugh]] would be directing the film, replacing Hughes, as well as Statham serving as a producer of the film,&lt;ref name=&quot;THR&quot; /&gt; while Stallone confirmed his involvement with the project, and that ''The Expendables 4'' was the same project as ''The Expendables: A Christmas Story'', previously reported as a [[Spin-off (media)|spin-off]] but in fact the film's working-title during development. Stallone further stated that production was set to commence in October of that year.&lt;ref name=&quot;:1&quot;&gt;{{cite web|url=https://screenrant.com/expendables-movie-spinoff-sylvester-stallone-filming-when-details/|work=[[Screen Rant]]|title=Sylvester Stallone Confirms Expendables Spinoff Starts Filming In October|author=Shirey, Paul|date=August 17, 2021|access-date=August 17, 2021|archive-date=August 17, 2021|archive-url=https://web.archive.org/web/20210817220331/https://screenrant.com/expendables-movie-spinoff-sylvester-stallone-filming-when-details/|url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;:2&quot;&gt;{{cite web|url=https://deadline.com/2021/07/expendables-4-sylvester-stallone-new-ring-1234807093/|work=[[Deadline Hollywood]]|title=Sylvester Stallone Teases Upcoming 'Expendables 4' With Barney New Ring|last=Choe|first=Brandon|date=July 31, 2021|access-date=August 30, 2021|archive-date=August 17, 2021|archive-url=https://web.archive.org/web/20210817233955/https://deadline.com/2021/07/expendables-4-sylvester-stallone-new-ring-1234807093/|url-status=live}}&lt;/ref&gt; According to [[Dolph Lundgren]], the film had a budget of $100 million.&lt;ref name=&quot;LundgrenInterview&quot;/&gt;
3318 In July 2018, [[Gregory Poirier]] announced his role as screenwriter.&lt;ref&gt;{{cite web|title=Expendables 4 Screenplay Spotted On Twitter|url=http://www.manlymovie.net/2018/10/expendables-4-screenplay-spotted-on-twitter.html|work=Manly Movie|date=October 22, 2018|author=The_Night_Rider|access-date=April 20, 2021|archive-date=July 11, 2023|archive-url=https://web.archive.org/web/20230711234046/https://www.manlymovie.net/2018/10/expendables-4-screenplay-spotted-on-twitter.html|url-status=live}}&lt;/ref&gt; Production was tentatively scheduled to begin by April 2019, though it wasn't until July of that year that Stallone announced that he was continuing to work on the script for the project.&lt;ref&gt;{{cite web|url=https://deadline.com/2018/05/the-expendables-4-sylvester-stallone-rambo-5-china-millenium-avi-lerner-dadi-1202395134/|title='The Expendables 4' Muscles Up: Sylvester Stallone Action Pic &amp; 'Rambo 5' Sell To China In Eight-Figure Co-Fi Deal|last=Wiseman|first=Andreas|date=May 21, 2018|work=[[Deadline Hollywood]]|access-date=April 20, 2021|archive-date=May 23,
2018|archive-url=https://web.archive.org/web/20180523011227/https://deadline.com/2018/05/the-expendables-4-sylvester-stallone-rambo-5-china-millenium-avi-lerner-dadi-1202395134/|url-status=live}}&lt;/ref&gt; The script was completed later that year, though negotiations with producers were ongoing.&lt;ref&gt;{{cite web |title=Randy Couture Interview on Final Kill |url=https://theactionelite.com/randy-couture-interview-on-final-kill/ |work=The Action Elite |last=Friel |first=Eoin |date=February 2020 |access-date=April 20, 2021 |archive-date=April 20, 2021 |archive-url=https://web.archive.org/web/20210420060659/https://theactionelite.com/randy-couture-interview-on-final-kill/ |url-status=live }}&lt;/ref&gt; In August 2021, it was announced that [[Spenser Cohen]] wrote the most recent draft of the script with [[Max D. Adams|Max Adams]], from a story by Cohen.&lt;ref name=&quot;THR&quot; /&gt;
Over the years, multiple actors reported interest or claimed they were approached to star in a fourth installment. In March 2014, [[Pierce Brosnan]] stated that he had agreed with producer [[Avi Lerner]] to star in a fourth installment.&lt;ref&gt;{{cite web|url=http://www.digitalspy.co.uk/movies/news/a560643/pierce-brosnan-to-appear-in-the-expendables-sequel.html/|title=Pierce Brosnan to appear in The Expendables sequel|first=Simon|last=Reynolds|work=Digital Spy|date=March 27, 2014|access-date=April 20, 2021|archive-date=September 24, 2015|archive-url=https://web.archive.org/web/20150924123205/http://www.digitalspy.co.uk/movies/news/a560643/pierce-brosnan-to-appear-in-the-expendables-sequel.html|url-status=live}}&lt;/ref&gt; By April of the same year, Sylvester Stallone revealed his first choice for the villain was [[Jack Nicholson]], while mentioning his interest in convincing [[Clint Eastwood]] to join the production.&lt;ref&gt;{{cite web|url=http://www.expendablespremiere.com/news/awesome-audio-interview-with-sly-stallone-reveals-jack-nicholson-is-his-first-choice-as-the-villain-for-expendables-4-also-talks-creed-and-rambo-5-new-mad-max-fury-road-trailer|title=Awesome audio interview with Sly Stallone, reveals Jack Nicholson is his first choice as the villain for Expendables 4 also talks Creed and Rambo 5 + New Mad Max Fury Road trailer|work=Expendables Premiere|access-date=April 20, 2021|archive-date=March 19, 2016|archive-url=https://web.archive.org/web/20160319220310/http://www.expendablespremiere.com/news/awesome-audio-interview-with-sly-stallone-reveals-jack-nicholson-is-his-first-choice-as-the-villain-for-expendables-4-also-talks-creed-and-rambo-5-new-mad-max-fury-road-trailer|url-status=dead}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://www.ign.com/articles/2014/04/16/stallone-wants-nicholson-and-eastwood-for-the-expendables|work=IGN|title=Stallone Wants Nicholson and Eastwood For The Expendables|author=Singer, Leigh|date=August 14, 2014|access-date=August 13, 2022|archive-date=October 8, 2021|archive-url=https://web.archive.org/web/20211008190827/https://www.ign.com/articles/2014/04/16/stallone-wants-nicholson-and-eastwood-for-the-expendables|url-status=live}}&lt;/ref&gt; According to [[Jackie Chan]], he was approached to star in the film by Stallone but was unsure about having limited screen time in an ensemble film, which is why he had turned down roles in the previous two entries.&lt;ref&gt;{{cite web |url=https://www.denofgeek.com/movies/jackie-chan-interview-chinese-zodiac-and-retiring/?amp |work=[[Den of Geek]] |title=Jackie Chan interview: Chinese Zodiac and retiring |last=Lambie |first=Ryan |date=July 27, 2014 |access-date=July 1, 2022 |archive-date=August 14, 2022 |archive-url=https://web.archive.org/web/20220814072954/https://www.denofgeek.com/movies/jackie-chan-interview-chinese-zodiac-and-retiring/?amp |url-status=live }}&lt;/ref&gt; [[Dwayne Johnson]] publicly expressed interest to play a villainous character in a new installment during a 2014 Q&amp;A&lt;ref&gt;{{cite web |url=https://www.cinemablend.com/new/Rock-Wants-Take-Down-Expendables-Other-Things-Learned-From-His-AMA-66420.html |work=[[CinemaBlend]] |title=The Rock Wants To Take Down The Expendables And Other Things Learned From His AMA |last=Reyes |first=Mike |date=July 27, 2014 |access-date=July 1, 2022 |archive-date=July 1, 2022 |archive-url=https://web.archive.org/web/20220701133503/https://www.cinemablend.com/new/Rock-Wants-Take-Down-Expendables-Other-Things-Learned-From-His-AMA-66420.html |url-status=live }}&lt;/ref&gt; and reiterated his interest to join the series in 2022.&lt;ref&gt;{{cite web |url=https://screenrant.com/dwayne-johnson-expendables-interview-gi-joe-2/ |work=[[Screen Rant]] |title=Dwayne Johnson Interview: The Rock Wants to Join The Expendables 4 Cast |last=Kendrick |first=Ben |date=February 15, 2022 |access-date=July 1, 2022 |archive-date=July 1, 2022 |archive-url=https://web.archive.org/web/20220701132000/https://screenrant.com/dwayne-johnson-expendables-interview-gi-joe-2/ |url-status=live }}&lt;/ref&gt; In May 2015, [[Hulk Hogan]] claimed that he was in talks with Stallone to play the film's main villain.&lt;ref name=&quot;Hogan&quot;/&gt;
However, none of the actors mentioned above were subsequently cast, and the cast was announced to include returning stars Sylvester Stallone, Jason Statham, [[Randy Couture]], and Dolph Lundgren, while new cast members were revealed to be [[Eddie Hall]], [[50 Cent|Curtis &quot;50 Cent&quot; Jackson]], [[Megan Fox]], [[Tony Jaa]], [[Andy García]], Sheila Shah, [[Jacob Scipio]] and [[Levy Tran]],&lt;ref name=&quot;THR&quot; /&gt;&lt;ref name=&quot;OctoberCast&quot; /&gt; with [[Iko Uwais]] playing the villain.&lt;ref name=&quot;Uwais&quot; /&gt;
In August 2021, it was stated that [[principal photography]] would begin in October.&lt;ref name=&quot;THR&quot;&gt;{{cite web|url=https://www.hollywoodreporter.com/movies/movie-news/jason-statham-sylvester-stallone-50-cent-megan-fox-new-expendables-movie-cast-1235004992/|work=[[The Hollywood Reporter]]|title=Jason Statham, Sylvester Stallone, 50 Cent, Megan Fox to Star in New 'The Expendables' Movie for Lionsgate (Exclusive)|first=Borys|last=Kit|date=August 30, 2021|access-date=August 30, 2021|archive-date=August 30, 2021|archive-url=https://web.archive.org/web/20210830160338/https://www.hollywoodreporter.com/movies/movie-news/jason-statham-sylvester-stallone-50-cent-megan-fox-new-expendables-movie-cast-1235004992/|url-status=live}}&lt;/ref&gt; Filming officially commenced on September 29, 2021.&lt;ref&gt;{{cite web|url=https://comicbook.com/movies/news/the-expendables-4-sylvester-stallone-and-jason-statham-share-first-set-photo/|work=[[ComicBook.com]]|title=The Expendables 4: Sylvester Stallone and Jason Statham Share First Set Photo|last=Jirak|first=Jamie|date=October 7, 2021|access-date=October 7, 2021|archive-date=October 9, 2021|archive-url=https://web.archive.org/web/20211009023351/https://comicbook.com/movies/news/the-expendables-4-sylvester-stallone-and-jason-statham-share-first-set-photo/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://screenrant.com/expendables-4-megan-fox-costume-set-image/|work=[[Screen Rant]]|title=Expendables 4 Set Photos Reveal First Look At Megan Fox's Costume|last=Klein|first=Brennan|date=October 7, 2021|access-date=October 7, 2021|archive-date=November 11, 2021|archive-url=https://web.archive.org/web/20211111205455/https://screenrant.com/expendables-4-megan-fox-costume-set-image/|url-status=live}}&lt;/ref&gt; In October 2021, Stallone announced on social media that he had finished filming his scenes for the movie.&lt;ref name=&quot;Devore&quot;/&gt; The film was shot in London and Bulgaria and the [[Jackie Chan Stunt Team]] handled stunt choreography.&lt;ref&gt;{{cite web |url=https://www.takt-magazin.de/magazin/stuntman-mike-moeller-aus-bad-lobenstein-kommt-ins-ganz-grosse-kino_312206 |title=Stuntman Mike Möller aus Bad Lobenstein kommt ins ganz große Kino |website=Takt Magazin |first=Lutz |last=Granert |date=August 19, 2022 |access-date=August 19, 2022 |language=German |archive-date=August 19, 2022 |archive-url=https://web.archive.org/web/20220819085000/https://www.takt-magazin.de/magazin/stuntman-mike-moeller-aus-bad-lobenstein-kommt-ins-ganz-grosse-kino_312206 |url-status=live }}&lt;/ref&gt; In November 2021, production was shot in [[Greece]], including the city of [[Thessaloniki]].&lt;ref&gt;{{cite web |url=https://deadline.com/2021/11/expendables-4-two-crew-members-injured-during-production-greece-1234875889/ |title='The Expendables 4': Two Crew Members Injured During Production In Europe |website=[[Deadline Hollywood]] |first=Andreas |last=Wiseman |date=November 17, 2021 |access-date=April 13, 2022 |archive-date=April 13, 2022 |archive-url=https://web.archive.org/web/20220413173735/https://deadline.com/2021/11/expendables-4-two-crew-members-injured-during-production-greece-1234875889/ |url-status=live }}&lt;/ref&gt; Members of the [[Greek Armed Forces]] were used as [[extra (acting)|extras]] and alleged they were not compensated for their month-long official deployment, despite working overtime.&lt;ref&gt;{{cite web |url=https://greekcitytimes.com/2022/04/13/expendables-4-refuses/?amp |title=Expendables 4 refuses to compensate Greek soldiers for acting in big budget film |website=Greek City Times |first=Athens |last=Bureau |date=April 13, 2022 |access-date=April 15, 2022 |archive-date=April 14, 2022 |archive-url=https://web.archive.org/web/20220414175118/https://greekcitytimes.com/2022/04/13/expendables-4-refuses/?amp |url-status=live }}&lt;/ref&gt; On December 3, 2021, Tony Jaa confirmed filming had wrapped.&lt;ref&gt;{{cite web |url=https://www.facebook.com/photo/?fbid=479401013548070&amp;set=a.220112362810271 |title=Thats a wrap for 'Expendables 4' |first=Tony |last=Jaa |website=[[Facebook]] |date=December 3, 2021 |access-date=June 10, 2022 |archive-date=June 6, 2022 |archive-url=https://web.archive.org/web/20220606175019/https://www.facebook.com/photo/?fbid=479401013548070&amp;set=a.220112362810271 |url-status=live }}&lt;/ref&gt;
''Expend4bles'' was released in mainland China on September 15, 2023, and was theatrically released by [[Lionsgate Films]] on September 22, 2023.&lt;ref&gt;{{cite web |url=https://deadline.com/2022/10/expendables-4-sebastian-maniscalco-about-my-father-release-dates-1235137654/ |title=Lionsgate's 'Expendables 4', Sebastian Maniscalco's 'About My Father' &amp; Adele Lim Comedy Get 2023 Release Dates |first=Anthony |last=D'Alessandro |work=[[Deadline Hollywood|Deadline]] |date=October 6, 2022 |access-date=October 7, 2022 |archive-date=October 6, 2022 |archive-url=https://web.archive.org/web/20221006224340/https://deadline.com/2022/10/expendables-4-sebastian-maniscalco-about-my-father-release-dates-1235137654/ |url-status=live }}&lt;/ref&gt; The film was originally scheduled to be released in 2022.&lt;ref&gt;{{cite web|url=https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video|title=Expendables 4: Sylvester Stallone Looks Thoroughly Badass In New Set Video|first=Erik|last=Swann|work=[[CinemaBlend]] |date=October 12, 2021|access-date=October 14, 2021|url-status=live|archive-date=October 14, 2021|archive-url=https://web.archive.org/web/20211014065910/https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video}}&lt;/ref&gt;
In April 2022, the first [[standee]] posters for the film were unveiled at [[CinemaCon]].&lt;ref&gt;{{cite web |url=https://comicbook.com/movies/amp/news/the-expendables-4-cast-poster-revealed/ |title=The Expendables 4 First Poster Revealed |first=Jamie |last=Lovett |work=[[ComicBook.com]] |date=April 25, 2022 |access-date=April 25, 2022 |archive-date=April 25, 2022 |archive-url=https://web.archive.org/web/20220425143329/https://comicbook.com/movies/amp/news/the-expendables-4-cast-poster-revealed/ |url-status=live }}&lt;/ref&gt; The first teaser trailer footage was released exclusively to the attendees at Lionsgate's presentation at the convention.&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/movies/movie-news/lionsgate-cinemacon-john-wick-4-expendables-1235137673/|work=The Hollywood Reporter|title='John Wick: Chapter 4' Footage Closes Out CinemaCon With a Bang|last=Gardner|first=Chris|date=April 28, 2022|access-date=April 28, 2022|archive-date=April 28, 2022|archive-url=https://web.archive.org/web/20220428223706/https://www.hollywoodreporter.com/movies/movie-news/lionsgate-cinemacon-john-wick-4-expendables-1235137673/|url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://cosmicbook.news/expendables-4-cinemacon-footage-sylvester-stallone|work=Cosmic Book News|title='The Expendables 4' CinemaCon Footage Shows Off Sylvester Stallone|last=McGloin|first=Matt|date=April 28, 2022|access-date=April 28, 2022|archive-date=May 20, 2022|archive-url=https://web.archive.org/web/20220520131710/https://cosmicbook.news/expendables-4-cinemacon-footage-sylvester-stallone|url-status=live}}&lt;/ref&gt; On June 6, 2023, the first poster for the film was released, with the first trailer releasing a day later.&lt;ref&gt;{{cite web|url=https://collider.com/the-expendables-4-poster/|work=Collider|title=New 'The Expendables 4' Poster Promises &quot;They'll Die When They're Dead&quot;|last=Echebiri|first=Makuochi|date=June 6, 2023|access-date=June 7, 2023|archive-date=June 7, 2023|archive-url=https://web.archive.org/web/20230607010432/https://collider.com/the-expendables-4-poster/|url-status=live}}&lt;/ref&gt; A second trailer advertising the film's R rating was released August 23, 2023.&lt;ref&gt;{{cite web|url= https://comicbook.com/movies/amp/news/expendables-4-new-trailer-sylvester-stallone-jason-statham/ |work=Comicbook.com|title= Expendables 4 Gets New Trailer |last=Brail|first=Nathaniel|date=August 23, 2023|access-date=August 31, 2023}}&lt;/ref&gt;
| [[Sylvester Stallone]]
| {{sort|Stallone|[[Sylvester Stallone]]}}
In June 2019, Walton appeared in the [[HBO]] television series ''[[Euphoria (American TV series)|Euphoria]]'' as Ashtray,&lt;ref name=&quot;Euphoria&quot;&gt;{{cite web |url=https://www.wmagazine.com/story/euphorias-child-drug-dealer-javon-walton-boxer |title=Javon Walton, Euphoria's Child Drug Dealer, Is Actually a Champion Boxer |website=[[W (magazine)|W Magazine]] |first=Brooke |last=Marine |date=June 30, 2019 |access-date=May 25, 2021}}&lt;/ref&gt; before landing a main role in the [[Amazon Prime Video]] streaming series ''[[Utopia (2020 TV series)|Utopia]]'' (2020) as Grant Bishop.&lt;ref name=&quot;Utopia&quot;&gt;{{cite web|url=https://deadline.com/2019/03/utopia-javon-wanna-walton-co-star-amazon-series-1202584849/|title='Utopia': Javon 'Wanna' Walton To Star In Amazon Series|website=[[Deadline Hollywood]]|first1=Nellie|last1=Andreeva|first2=Denise|last2=Petski|date=March 29, 2019|access-date=May 25, 2021|archive-date=October 10, 2020|archive-url=https://web.archive.org/web/20201010210316/https://deadline.com/2019/03/utopia-javon-wanna-walton-co-star-amazon-series-1202584849/|url-status=live}}&lt;/ref&gt; In 2021, he appeared in the animated comedy horror film ''[[The Addams Family 2]]'' as [[Pugsley Addams]].&lt;ref name=&quot;Hader&quot;&gt;{{Cite web |url=https://deadline.com/2020/10/the-addams-family-2-bill-hader-javon-wanna-walton-join-voice-cast-1234593614/ |title='The Addams Family 2': Bill Hader &amp; Javon &quot;Wanna&quot; Walton Join Voice Cast |first=Anthony |last=D'Alessandro |website=[[Deadline Hollywood]] |date=October 8, 2020 |archive-url=https://web.archive.org/web/20201008162155/https://deadline.com/2020/10/the-addams-family-2-bill-hader-javon-wanna-walton-join-voice-cast-1234593614/ |archive-date=October 8, 2020 |url-status=live |access-date=May 25, 2021}}&lt;/ref&gt; In 2022, he starred in the [[Netflix]] superhero television series ''[[The Umbrella Academy (TV series)|The Umbrella Academy]],'' and in the superhero thriller film ''[[Samaritan (film)|Samaritan]]'' as Sam Cleary.&lt;ref&gt;{{cite web |date= February 21, 2020 |last= N'Duka |first= Amanda |title= Sylvester Stallone Thriller 'Samaritan' Adds Jared Odrick &amp; Michael Aaron Milligan; Christian DelGrosso Joins 'Wally's Wonderland' - Film Briefs |url=
```

```
https://deadline.com/2020/02/sylvester-stallone-samaritan-jared-odrick-michael-aaron-milligan-christian-delgrosso-wallys-wonderland-film-briefs-1202862151/ |website= [[Deadline Hollywood]] |access-date=May 25, 2021 }}&lt;/ref&gt; In 2024, he starred in the streaming series ''[[Under the Bridge (TV series)|Under the Bridge]]''.&lt;ref&gt;{{cite web|url=https://deadline.com/2022/12/archie-panjabi-under-the-bridge-vritika-gupta-aiyana-goodfellow-cast-hulu-series-1235192709/|title=Archie Panjabi Joins 'Under the Bridge'; Vritika Gupta, Javon Walton &amp; Aiyana Goodfellow Also Cast In Hulu Series|website=Deadline Hollywood|first=Nellie|last=Andreeva|date=December 8, 2022|access-date=June 23, 2023}}&lt;/ref&gt;
3327 | ''[[Rocky (franchise)|Rocky]]''&lt;br/&gt;&lt;small&gt;([[Sylvester Stallone]])&lt;/small&gt;|| yes || no || no || '''''[[Rocky]]''''' (1976)&lt;br/&gt;Numerous sequels and spin-offs || no || no || Various || [[Rocky the Musical]]
3328 | ShortSummary    = {{small|'''Celebrities impersonated:''' John Humphrys, Sue MacGregor, Jim Naughtie (as Graham Norton), Alastair Stewart, Jack Straw, [[David Frost]], Loyd Grossman, Dalek, Charlotte Green, Captain Kirk (William Shatner), [[Scotty (Star Trek)|Mr. Scott]] ([[James Doohan]]), Robin Cook, Brian Perkins, [[Libby Purves]], Chris Tarrant, Henry Blofeld, Dustin Hoffman, Brian Sewell, Dale Winton, [[Bernard Manning]], Frank Bruno, John Major, [[Sylvester Stallone]], [[Peter O'Sullevan]], John Prescott, Ned Sherrin, Charlie Dimmock, [[Alan Titchmarsh]], Chris Eubank, Rolf Harris, Fred Trueman, Jerry Springer, Yogi Bear, Boo-Boo Bear, [[Sara Cox]], Des Lynam, Alan Bennett.}}
3329 | ShortSummary    = {{small|'''Celebrities impersonated:''' [[Glenn Campbell (broadcaster)|Glenn Campbell]], [[Alistair Darling]], Alex Salmond, [[Andy Murray]], Evan Davis, Jim Naughtie, [[Jeremy Clarkson]], [[Sylvester Stallone]], [[Arnold Schwarzenegger]], [[Mel Gibson]], Al Pacino, Nigel Farage, Cora Crawley (Elizabeth McGovern), Robert Crawley (Hugh Bonneville), Isobel Crawley (Penelope Wilton), Charles Carson (Jim Carter), Dowager Countess of Grantham (Maggie Smith), Lady Rose MacClare ([[Lily James]]), Ray Winstone, [[Vince Cable]], Jools Holland, [[Winifred Robinson]], [[Morrissey]], Neil Nunes, Helen Titchener ([[Louiza Patikas]]), Rob Titchener ([[Timothy Watson]]), Penelope Wilton, [[Alan Carr]], Alan Bennett, Ian McKellen, Patrick Stewart, [[David Beckham]], Maggie Smith, [[Sue MacGregor]], [[Saint Peter]], [[Thomas the Apostle]], [[Judas Iscariot]], David Cameron, Ed Miliband.}}
3330 | ShortSummary    = {{small|'''Celebrities impersonated:''' Neil Nunes, David Attenborough, Angela Eagle, John Humphrys, Sarah Montague, Donald Trump, Andrew Neil, Theresa May, [[Amber Rudd]], [[David Davis (British politician)|David Davis]], Boris Johnson, Neil Oliver, [[Recep Tayyip Erdoğan]], Jeremy Corbyn, [[Hugh Grant]], Sylvester Stallone, Trevor McDonald, Mo Farah, Roy Hodgson, Jim Naughtie, Huw Edwards, Laura Kuenssberg, [[Philip Hammond]], [[Sam Allardyce]], [[Pippa Middleton]], Queen Elizabeth II, Morgan Freeman, Al Pacino, Alan Bennett, Jon Snow, [[Taylor Swift]], Michael Gove, Diane Abbott, Nick Clegg, Tim Farron, Ozzy Osbourne, [[Christopher Walken]], Greg Dyke, Nick Robinson, Kay Burley, Ian McKellen, Michael Caine.}}
3331 DaVinci International Film Festival's 5th Edition announced its partnership with [[AMC Theatres]] at [[The Grove at Farmers Market]] and was held September 24-25, 2022.&lt;ref&gt;{{cite web|title=TREND Magazine|url=https://thetrendmag.com/media/davinci-international-film-festival-announces-its-5th-edition-to-be-held-at-the-grove-in-los-angeles-september-24th-25th/|website=thetrendmag.com}}&lt;/ref&gt; In addition to its regular programming, the festival introduced all-new panel events including ''Film [[Canada]]'' presented by industry leader, William F. White International&lt;ref&gt;{{cite web|title=NEWSWIRES|url=https://www.einnews.com/pr_news/586194792/davinci-international-film-festival-and-william-f-white-international-to-join-forces-for-all-new-film-canada-event/|website=einnews.com}}&lt;/ref&gt; and ''Digital Hollywood'', which invited actor and [[Executive Producer]], [[Aleks Paunovic]] and business partner, [[Neil Stevenson-Moore]] to discuss [[NFT]]'s, [[blockchain]], and how the new tech is disrupting Hollywood traditional financial systems.&lt;ref&gt;{{cite web|title=User Walls|url=https://www.userwalls.news/n/davinci-international-film-festival-set-debut-digital-hollywood-program-featuring-live-action-3640921/|website=userwalls.news}}&lt;/ref&gt; Founder and President, Chadwick Pelletier presented DIFF's first-ever Honorary Leo Award to ''Destination Angels'', a multi-media [[Jack Kerouac]] Centennial [[documentary]] event, directed by Daniel Lir and produced by [[Grammy Award]] winning musical artist, [[Dru DeCaro]].&lt;ref&gt;{{cite web|title=AP News|url=https://apnews.com/article/entertainment-film-festivals-jack-kerouac-d1cd4f458db88b7b0a4f446539c4cca9/|website=apnews.com}}&lt;/ref&gt;&lt;ref&gt;{{cite web|title=Destination Angeles: A Jack Kerouac Centennial |url=https://davincifilmfestival.com/jack-kerouac/|website=davincifilmfestival.com}}&lt;/ref&gt; Among some of the titles in competition were [[Sylvester Stallone]]'s ''MVP''. [[New Zealand]]'s''The Justice of Bunny King'' (Leo Award Winner&lt;ref&gt;{{cite web|title=2022 Program &amp; Results |url=https://davincifilmfestival.com/2022-program/|website=davincifilmfestival.com}}&lt;/ref&gt;), a [[Valerie Perrine]] documentary, and ''Tough Ain't Enough'', which documents two-time [[Academy Award]] winning producer, [[Albert S. Ruddy]] and his work on such films as [[Hogan's Heroes]], [[The Godfather]], and [[Million Dollar Baby]], among others.&lt;ref&gt;{{cite web|title=2022 Program &amp; Results |url=https://davincifilmfestival.com/2022-program/|website=davincifilmfestival.com}}&lt;/ref&gt;
3332 (957) [[S. E. Hinton]] · [[S. E. Rogie]] · [[S. R. Ranganathan]] · [[S. S. Kresge]] · [[S. T. Joshi]] · [[S.E.S. (group)]] · [[S.H.E]] · [[SMAP]] · [[Sa'd ibn Abi Waqqas]] · [[Saad Al-Salim Al-Sabah]] · [[Saadi Shirazi]] · [[Saadia Gaon]] · [[Sabah Al-Salim Al-Sabah]] · [[Sabbatai Zevi]] · [[Sable Starr]] · [[Sabri Brothers]] · [[Sacagawea]] · [[Sacha Baron Cohen]] · [[Sacha Guitry]] · [[Sacheverell Sitwell]] · [[Sachin Tendulkar]] · [[Sada Abe]] · [[Sadaharu Oh]] · [[Saddam Hussein]] · [[Sade (band)]] · [[Sadegh Hedayat]] · [[Sai On]] · [[Said bin Sultan]] · [[Said bin Taimur]] · [[Saif Ali Khan]] · [[Saif Saaeed Shaheen]] · [[Saif bin Sultan]] · [[Saigyō]] · [[Saigō Takamori]] · [[Sailor Jerry]] · [[Saina Nehwal]] · [[Saint Christopher]] · [[Saint Denis of Paris]] · [[Saint Dominic]] · [[Saint George]] · [[Saint Joseph]] · [[Saint Marinus]] · [[Saint Matthias]] · [[Saint Nicholas]] · [[Saint Patrick]] · [[Saint Peter]] · [[Saint Stephen]] · [[Saint Timothy]] · [[Saint-John Perse]] · [[Saints Cyril and Methodius]] · [[Sakamoto Ryōma]] · [[Sakichi Toyoda]] · [[Sal Mineo]] · [[Saladin]] · [[Salah Ahmed Ibrahim]] · [[Salah Jahin]] · [[Salah al-Din al-Bitar]] · [[Salim Khan]] · [[Sallust]] · [[Sally Field]] · [[Sally Jessy Raphael]] · [[Sally Mann]] · [[Sally Morgan (artist)]] · [[Sally Pearson]] · [[Sally Rand]] · [[Sally Ride]] · [[Sally Rooney]] · [[Sally Salisbury]] · [[Salma Hayek]] · [[Salman Khan]] · [[Salman Rushdie]] · [[Salman of Saudi Arabia]] · [[Salman the Persian]] · [[Saloma]] · [[Salt-N-Pepa]] · [[Salva Kiir Mayardit]] · [[Salvador Allende]] · [[Salvador Dalí]] · [[Salvador Sánchez Cerén]] · [[Salvadore Cammarano]] · [[Salvatore Quasimodo]] · [[Salvatore Riina]] · [[Sam &amp; Dave]] · [[Sam Bockarie]] · [[Sam Cooke]] · [[Sam Giancana]] · [[Sam Harris]] · [[Sam Hinds]] · [[Sam Houston]] · [[Sam Hui]] · [[Sam Kinison]] · [[Sam Mendes]] · [[Sam Nujoma]] · [[Sam Peckinpah]] · [[Sam Phillips]] · [[Sam Poo]] · [[Sam Rayburn]] · [[Sam Rockwell]] · [[Sam Shepard]] · [[Sam Snead]] · [[Sam Walton]] · [[Sam Warner]] · [[Sam Zolotow]] · [[Saman Khuda]] · [[Samantha Bee]] · [[Samantha Cristoforetti]] · [[Samantha Lewthwaite]] · [[Samantha Smith]] · [[Samart Payakaroon]] · [[Sami Yusuf]] · [[Sammy Baugh]] · [[Sammy Davis Jr.]] · [[Sammy Gravano]] · [[Samora Machel]] · [[Samora Yunis]] · [[Samori Ture]] · [[Samson Raphael Hirsch]] · [[Samudragupta]] · [[Samuel Adams]] · [[Samuel Baker]] · [[Samuel Barber]] · [[Samuel Beckett]] · [[Samuel Brannan]] · [[Samuel Butler (poet)]] · [[Samuel C. C. Ting]] · [[Samuel Coleridge-Taylor]] · [[Samuel Colt]] · [[Samuel Conway]] · [[Samuel Doe]] · [[Samuel Eilenberg]] · [[Samuel Eliot Morison]] · [[Samuel Eto'o]] · [[Samuel Flagg Bemis]] · [[Samuel Fuller]] · [[Samuel Goldwyn]] · [[Samuel Gompers]] · [[Samuel Hahnemann]] · [[Samuel Insull]] · [[Samuel Johnson]] · [[Samuel Keimer]] · [[Samuel Kleinschmidt]] · [[Samuel L. Jackson]] · [[Samuel Langley]] · [[Samuel Little]] · [[Samuel Morse]] · [[Samuel Orchart Beeton]] · [[Samuel P. Huntington]] · [[Samuel Pepys]] · [[Samuel Rawson Gardiner]] · [[Samuel Richardson]] · [[Samuel Rogers]] · [[Samuel Sachs]] · [[Samuel Samuel]] · [[Samuel Slater]] · [[Samuel Taylor Coleridge]] · [[Samuel Wallis]] · [[Samuel Wanjiru]] · [[Samuel de Champlain]] · [[Samuel of Bulgaria]] · [[Sanada Yukimura]] · [[Sanath Jayasuriya]] · [[Sancho III of Pamplona]] · [[Sancho VI of Navarre]] · [[Sancho VII of Navarre]] · [[Sandra Bullock]] · [[Sandra Day O'Connor]] · [[Sandra Faber]] · [[Sandra Perković]] · [[Sandra Schmirler]] · [[Sandro Botticelli]] · [[Sandro de América]] · [[Sandy Dennis]] · [[Sandy Gall]] · [[Sandy Koufax]] · [[Sanford B. Dole]] · [[Sanford I. Weill]] · [[Sang Nila Utama]] · [[Sani Abacha]] · [[Sanjay Leela Bhansali]] · [[Sanjeev Kumar]] · [[Sanmao (writer)]] · [[Santana (band)]] · [[Santiago Calatrava]] · [[Santiago Ramón y Cajal]] · [[Santō Kyōden]] · [[Sao Shwe Thaik]] · [[Saori Yoshida]] · [[Saparmurat Niyazov]] · [[Sappho]] · [[Sara Montiel]] · [[Sara Teasdale]] · [[Sarah Baartman]] · [[Sarah Bernhardt]] · [[Sarah Brightman]] · [[Sarah Churchill, Duchess of Marlborough]] · [[Sarah Geronimo]] · [[Sarah Jessica Parker]] · [[Sarah Josepha Hale]] · [[Sarah Kane]] · [[Sarah Koenig]] · [[Sarah Michelle Gellar]] · [[Sarah Palin]] · [[Sarah Siddons]] · [[Sarah Silverman]] · [[Sarah Vaughan]] · [[Sarah Winnemucca]] · [[Saraha]] · [[Sarantuya]] · [[Sargon II]] · [[Sargon of Akkad]] · [[Saritha]] · [[Saroj Khan]] · [[Sarvepalli Radhakrishnan]] · [[Sasha Chorny]] · [[Sasha Grey]] · [[Sassoon family]] · [[Sataspes]] · [[Satchel Paige]] · [[Sathya Sai Baba]] · [[Satoru Iwata]] · [[Satoshi Nakamoto]] · [[Satoshi Tajiri]] · [[Satyajit Ray]] · [[Satyendra Nath Bose]] · [[Saud of Saudi Arabia]] · [[Saul Alinsky]] · [[Saul Bass]] · [[Saul Bellow]] · [[Saul Kripke]] · [[Saul Perlmutter]] · [[Saunders Mac Lane]] · [[Savion Glover]] · [[Sawao Katō]] · [[Sayed Darwish]] · [[Sayyid Qutb]] · [[Scarlett Johansson]] · [[Schlitzie]] · [[Scipio Africanus]] · [[Scopas]] · [[Scorpions (band)]] · [[Scott Carpenter]] · [[Scott Cutlip]] · [[Scott Hamilton (figure skater)]] · [[Scott Joplin]] · [[Scott Kelly (astronaut)]] · [[Scott McCloud]] · [[Scott Moir]] · [[Scott Morrison]] · [[Scott Rudin]] · [[Scotty Bowman]] · [[Scylax of Caryanda]] · [[Seamus Heaney]] · [[Sean Combs]] · [[Sean Connery]] · [[Sean Hannity]] · [[Sean Parker]] · [[Sean Paul]] · [[Sean Penn]] · [[Sebastian Cabot (explorer)]] · [[Sebastian Coe]] · [[Sebastian Faulks]] · [[Sebastian Vettel]] · [[Sebastiano Serlio]] · [[Sebastián Garcilaso de la Vega y Vargas]] · [[Sebastián Piñera]] · [[Sebastião José de Carvalho e Melo, 1st Marquis of Pombal]] · [[Sebastião Salgado]] · [[Sebhat Ephrem]] · [[Seen (artist)]] · [[Seewoosagur Ramgoolam]] · [[Sei Shōnagon]] · [[Seiichi Morimura]] · [[Seijun Suzuki]] · [[Seiu Ito]] · [[Sejong the Great]] · [[Seka Gadiyev]] · [[Seki Takakazu]] · [[Seku Amadu]] · [[Selena]] · [[Selena Gomez]] · [[Seleucus I Nicator]] · [[Selim I]] · [[Selim III]] · [[Selma Lagerlöf]] · [[Semion Mogilevich]] · [[Semonides of Amorgos]] · [[Semyon Budyonny]] · [[Semyon Dezhnev]] · [[Semyon Timoshenko]] · [[Seneca the Younger]] · [[Senenmut]] · [[Sennacherib]] · [[Senusret I]] · [[Senusret II]] · [[Senusret III]] · [[Seo Taiji and Boys]] · [[Seonjo of Joseon]] · [[Sepp Blatter]] · [[Sepp Dietrich]] · [[Septimius Severus]] · [[Sepultura]] · [[Sequoyah]] · [[Serena Williams]] · [[Seretse Khama]] · [[Serge Blanco]] · [[Serge Gainsbourg]] · [[Serge Haroche]] · [[Serge Koussevitzky]] · [[Serge Lang]] · [[Sergei Belov]] · [[Sergei Bodrov Jr.]] · [[Sergei Bondarchuk]] · [[Sergei Chavain]] · [[Sergei Diaghilev]] · [[Sergei Eisenstein]] · [[Sergei Kirov]] · [[Sergei Korolev]] · [[Sergei Lemeshev]] · [[Sergei Lukyanenko]] · [[Sergei Makarov (ice hockey)]] · [[Sergei Mozyakin]] · [[Sergei Parajanov]] · [[Sergei Polunin]] · [[Sergei Prokofiev]] · [[Sergei Rachmaninoff]] · [[Sergei Starostin]] · [[Sergei Witte]] · [[Sergei Yesenin]] · [[Sergey Brin]] · [[Sergey Bubka]] · [[Sergey Ivanovich Lomanov]] · [[Sergey Mikhalkov]] · [[Sergey Nechayev]] · [[Sergey Prokudin-Gorsky]] · [[Sergey Solovyov (historian)]] · [[Sergio Leone]] · [[Sergio Moro]] · [[Sergiu Celibidache]] · [[Seru Epenisa Cakobau]] · [[Serzh Sargsyan]] · [[Sesshū Tōyō]] · [[Sessue Hayakawa]] · [[Seth Bullock]] · [[Seth MacFarlane]] · [[Seth Meyers]] · [[Seth Rogen]] · [[Seti I]] · [[Setsuko Hara]] · [[Setthathirath]] · [[Seung-Hui Cho]] · [[Seve Ballesteros]] · [[Severina (singer)]] · [[Severus Alexander]] · [[Sex Pistols]] · [[Sextus Pompey]] · [[Seymour Cray]] · [[Seymour Hersh]] · [[Seymour Martin Lipset]] · [[Sezen Aksu]] · [[Seán MacBride]] · [[Seán O'Casey]] · [[Shaaban bin Robert]] · [[Shabana Azmi]] · [[Shadi Abdel Salam]] · [[Shadia]] · [[Shah Jahan]] · [[Shah Rukh Khan]] · [[Shahram Nazeri]] · [[Shahu I]] · [[Shahzoda]] · [[Shaka]] · [[Shakib Al Hasan]] · [[Shakira]] · [[Shamshi-Adad I]] · [[Shanakdakhete]] · [[Shane Dawson]] · [[Shane Smith (journalist)]] · [[Shane Warne]] · [[Shang Yang]] · [[Shania Twain]] · [[Shankar-Ehsaan-Loy]] · [[Shannon Larratt]] · [[Shannon Miller]] · [[Shapur I]] · [[Shapur II]] · [[Shaquille O'Neal]] · [[Sharaku]] · [[Sharifah Aini]] · [[Sharmila Tagore]] · [[Sharon Olds]] · [[Sharon Stone]] · [[Sharon Tate]] · [[Shashanka]] · [[Shaun Murray]] · [[Shaun White]] · [[Shavkat Mirziyoyev]] · [[Shawn Fanning]] · [[Shaykh Tusi]] · [[Shehu Shagari]] · [[Sheikh Hamdullah]] · [[Sheikh Hasina]] · [[Sheikh Mujibur Rahman]] · [[Sheila Nevins]] · [[Shekhar Kapur]] · [[Shel Dorf]] · [[Shel Silverstein]] · [[Sheldon Adelson]] · [[Sheldon Lee Glashow]] · [[Shelley Berman]] · [[Shelley Winters]] · [[Shelly-Ann Fraser-Pryce]] · [[Shen Buhai]] · [[Shen Congwen]] · [[Shen Dao]] · [[Shen Kuo]] · [[Shen Zhou]] · [[Shepard Fairey]] · [[Sher Bahadur Deuba]] · [[Sher Shah Suri]] · [[Shereen Bhan]] · [[Sheridan Le Fanu]] · [[Sherilyn Fenn]] · [[Sherry Ortner]] · [[Sherwood Anderson]] · [[Sherwood Washburn]] · [[Sheryl Sandberg]] · [[Sheryl Swoopes]] · [[Sheshi]] · [[Shi Lang]] · [[Shi Nai'an]] · [[Shibata Zeshin]] · [[Shibusawa Eiichi]] · [[Shigeru Ban]] · [[Shigeru Miyamoto]] · [[Shigeru Yoshida]] · [[Shilpa Shetty]] · [[Shimon Peres]] · [[Shin Jung-hyeon]] · [[Shin Kyung-sook]] · [[Shin Sang-ok]] · [[Shin'ichirō Tomonaga]] · [[Shinee]] · [[Shingo Kunieda]] · [[Shinsui Itō]] · [[Shintaro Ishihara]] · [[Shintaro Katsu]] · [[Shinzo Abe]] · [[Shirin Ebadi]] · [[Shirley Bassey]] · [[Shirley Booth]] · [[Shirley Chisholm]] · [[Shirley Jackson]] · [[Shirley Jones]] · [[Shirley MacLaine]] · [[Shirley Strickland]] · [[Shirley Temple]] · [[Shirō Ishii]] · [[Shitao]] · [[Shivaji]] · [[Shivaji Satam]] · [[Shivnarine Chanderpaul]] · [[Shizuka Arakawa]] · [[Shlomo Carlebach (musician)]] · [[Shmuel Yosef Agnon]] · [[Shneur Zalman of Liadi]] · [[Shoeless Joe Jackson]] · [[Shoghi Effendi]] · [[Shoichiro Toyoda]] · [[Shoko Asahara]] · [[Sholem Aleichem]] · [[Shonda Rhimes]] · [[Shong Lue Yang]] · [[Shoqan Walikhanov]] · [[Shosheng I]] · [[Shosheng II]] · [[Shreya Ghoshal]] · [[Shudraka]] · [[Shuji Nakamura]] · [[Shukri al-Quwatli]] · [[Shulamith Firestone]] · [[Shunpei Yamazaki]] · [[Shunryū Suzuki]] · [[Shunzhi Emperor]] · [[Shutruk-Nakhunte]] · [[Shyama Shastri]] · [[Shō Hashi]] · [[Shō Nei]] · [[Shō Tai]] · [[Shōji Hamada]] · [[Shūsaku Endō]] · [[Siad Barre]] · [[Siaka Stevens]] · [[Sid Caesar]] · [[Sid Meier]] · [[Sid Waddell]] · [[Sidharth Shukla]] · [[Sidney Bechet]] · [[Sidney Crosby]] · [[Sidney Lumet]] · [[Sidney Mintz]] · [[Sidney Nolan]] · [[Sidney Poitier]] · [[Sidney Reilly]] · [[Sidney Rigdon]] · [[Sidney Sheldon]] · [[Siegfried &amp; Roy]] · [[Siegfried Sassoon]] · [[Sigfrid Edström]] · [[Sigismund I the Old]] · [[Sigismund II Augustus]] · [[Sigismund III Vasa]] · [[Sigismund, Holy Roman Emperor]] · [[Sigmund Freud]] · [[Sigmund Jähn]] · [[Sigrid Undset]] · [[Sigur Rós]] · [[Silas]] · [[Silvana Armenulić]] · [[Silvana Mangano]] · [[Silver Horn]] · [[Silvia Neid]] · [[Silvio Berlusconi]] · [[Silvio Santos]] · [[Sima Qian]] · [[Sima Xiangru]] ·
```

```
[[Sima Yi]] • [[Simeon I of Bulgaria]] • [[Simin Daneshvar]] • [[Simmias (explorer)]] • [[Simo Häyhä]] • [[Simon &amp; Garfunkel]] • [[Simon Armitage]] • [[Simon Baron-Cohen]] • [[Simon Blackburn]] • [[Simon Cowell]] • [[Simon Fuller]] • [[Simon Kuznets]] • [[Simon Leys]] • [[Simon Magus]] • [[Simon Marius]] • [[Simon Newcomb]] • [[Simon Rattle]] • [[Simon Reynolds]] • [[Simon Schama]] • [[Simon Wiesenthal]] • [[Simon bar Kokhba]] • [[Simon de Montfort, 6th Earl of Leicester]] • [[Simon the Zealot]] • [[Simon van der Meer]] • [[Simone Biles]] • [[Simone Bittencourt de Oliveira]] • [[Simone Martini]] • [[Simone Niggli-Luder]] • [[Simone Signoret]] • [[Simone Weil]] • [[Simone de Beauvoir]] • [[Simonetta Vespucci]] • [[Simonides of Ceos]] • [[Siméon Denis Poisson]] • [[Simón Bolívar]] • [[Sin Yun-bok]] • [[Sinclair Lewis]] • [[Sinn Sisamouth]] • [[Siouxsie and the Banshees]] • [[Sir Arthur Harris, 1st Baronet]] • [[Sir George Stokes, 1st Baronet]] • [[Sirhan Sirhan]] • [[Sirimavo Bandaranaike]] • [[Sirusho]] • [[Sisavang Vatthana]] • [[Sisavang Vong]] • [[Sissy Spacek]] • [[Sister Ping]] • [[Sister Rosetta Tharpe]] • [[Sitara Devi]] • [[Siti Nurhaliza]] • [[Siti binti Saad]] • [[Sitiveni Rabuka]] • [[Sitting Bull]] • [[Sixto Durán Ballén]] • [[Sjokz]] • [[Sjón]] • [[Skanderbeg]] • [[Skepta]] • [[Skip Bayless]] • [[Skrillex]] • [[Slamet Rahardjo]] • [[Slavoj Žižek]] • [[Slayer]] • [[Slick Rick]] • [[Slim Dusty]] • [[Slobodan Milošević]] • [[Slobodan Praljak]] • [[Sly and the Family Stone]] • [[Smedley Butler]] • [[Smita Patil]] • [[Smosh]] • [[Smyrna Trio]] • [[Sneferu]] • [[Snooki]] • [[Snoop Dogg]] • [[Snorri Sturluson]] • [[Soad Hosny]] • [[Sobekneferu]] • [[Sobhuza II]] • [[Socrates]] • [[Socrates of Constantinople]] • [[Soda Stereo]] • [[Sofia Coppola]] • [[Sofia Rotaru]] • [[Sofonisba Anguissola]] • [[Sohrab Sepehri]] • [[Soichiro Honda]] • [[Sojourner Truth]] • [[Sokei-an]] • [[Sol LeWitt]] • [[Solomon Asch]] • [[Solomon Mamaloni]] • [[Solomon R. Guggenheim]] • [[Solon]] • [[Sombat Metanee]] • [[Somdej Toh]] • [[Son Ye-jin]] • [[Song Yu]] • [[Songtsen Gampo]] • [[Sonia Gandhi]] • [[Sonic Youth]] • [[Sonja Barend]] • [[Sonja Henie]] • [[Sonni Ali]] • [[Sonny Barger]] • [[Sonny Rollins]] • [[Sono Osato]] • [[Sonu Nigam]] • [[Sony Lab'ou Tansi]] • [[Sonya Golden Hand]] • [[Soong Ching-ling]] • [[Soong Mei-ling]] • [[Sophia Alekseyevna of Russia]] • [[Sophia Loren]] • [[Sophia of Hanover]] • [[Sophie Blanchard]] • [[Sophie Germain]] • [[Sophie Kurys]] • [[Sophie Scholl]] • [[Sophie, Countess of Bar]] • [[Sophocles]] • [[Sorley MacLean]] • [[Souleymane Cissé (film director)]] • [[Souls (band)]] • [[Soundarya]] • [[Souphanouvong]] • [[Sourav Ganguly]] • [[Southern All Stars]] • [[Sozomen]] • [[Spartacus]] • [[Spede Pasanen]] • [[Spencer Compton, 1st Earl of Wilmington]] • [[Spencer Perceval]] • [[Spencer Tracy]] • [[Spice Girls]] • [[Spike Lee]] • [[Spike Milligan]] • [[Spiro Agnew]] • [[Spyros Kyprianou]] • [[Srečko Kosovel]] • [[Sri Aurobindo]] • [[Sridevi]] • [[Srinivasa Ramanujan]] • [[Stalking Cat]] • [[Stamford Raffles]] • [[Stan Brakhage]] • [[Stan Laurel]] • [[Stan Lee]] • [[Stan Musial]] • [[Stan and Jan Berenstain]] • [[Stanislas-Marie Maillard]] • [[Stanislav Petrov]] • [[Stanislaw Ulam]] • [[Stanisław August Poniatowski]] • [[Stanisław Koniecpolski]] • [[Stanisław Lem]] • [[Stanisław Leszczyński]] • [[Stanisława Przybyszewska]] • [[Stanley Baldwin]] • [[Stanley Bruce]] • [[Stanley Cohen (sociologist)]] • [[Stanley Donen]] • [[Stanley Finch]] • [[Stanley Ho]] • [[Stanley Holloway]] • [[Stanley Kramer]] • [[Stanley Kubrick]] • [[Stanley Matthews]] • [[Stanley Middleton]] • [[Stanley Milgram]] • [[Stanley Rous]] • [[Stanley Schachter]] • [[Stanley Williams]] • [[Stanton T. Friedman]] • [[Steele Rudd]] • [[Steely Dan]] • [[Steen Steensen Blicher]] • [[Stefan Banach]] • [[Stefan Dušan]] • [[Stefan Edberg]] • [[Stefan Nemanja]] • [[Stefan Raab]] • [[Stefan Yavorsky]] • [[Stefan Zweig]] • [[Stefan the First-Crowned]] • [[Stefanie Sun]] • [[Steffi Graf]] • [[Stefka Kostadinova]] • [[Stefán Karl Stefánsson]] • [[Stella Adler]] • [[Stendhal]] • [[Stenka Razin]] • [[Stepan Bandera]] • [[Steph Davis]] • [[Stephanie Flanders]] • [[Stephanie Kwolek]] • [[Stephanie Okereke Linus]] • [[Stephanie St. Clair]] • [[Stephen A. Douglas]] • [[Stephen Báthory]] • [[Stephen Colbert]] • [[Stephen Cook]] • [[Stephen Crane]] • [[Stephen Curry]] • [[Stephen Decatur]] • [[Stephen E. Ambrose]] • [[Stephen Foster]] • [[Stephen Fry]] • [[Stephen Girard]] • [[Stephen Glass]] • [[Stephen Gray (scientist)]] • [[Stephen Hales]] • [[Stephen Harper]] • [[Stephen Hawking]] • [[Stephen Hendry]] • [[Stephen Hillenburg]] • [[Stephen I of Hungary]] • [[Stephen Jay Gould]] • [[Stephen King]] • [[Stephen Mather]] • [[Stephen McNallen]] • [[Stephen Paddock]] • [[Stephen Rodefer]] • [[Stephen Schwartz (composer)]] • [[Stephen Sondheim]] • [[Stephen Spender]] • [[Stephen Van Rensselaer]] • [[Stephen the Great]] • [[Stephen, King of England]] • [[Stephenie Meyer]] • [[Steppenwolf (band)]] • [[Stesichorus]] • [[Steve Allen]] • [[Steve Bannon]] • [[Steve Biko]] • [[Steve Brodie (bridge jumper)]] • [[Steve Burns]] • [[Steve Caballero]] • [[Steve Carell]] • [[Steve Cauthen]] • [[Steve Cohen (businessman)]] • [[Steve Ditko]] • [[Steve Fossett]] • [[Steve Gerber]] • [[Steve Harvey]] • [[Steve Irwin]] • [[Steve Jobs]] • [[Steve Martin]] • [[Steve McQueen]] • [[Steve Nash]] • [[Steve Redgrave]] • [[Steve Reeves]] • [[Steve Reich]] • [[Steve Ritchie (pinball designer)]] • [[Steve Shirley]] • [[Steve Smith (cricketer)]] • [[Steve Vai]] • [[Steve Waugh]] • [[Steve Wilkos]] • [[Steve Wozniak]] • [[Steve Wynn]] • [[Steve Yzerman]] • [[Steve-O]] • [[Steven Bradbury]] • [[Steven Chu]] • [[Steven Curtis Chapman]] • [[Steven Kanumba]] • [[Steven Pinker]] • [[Steven Seagal]] • [[Steven Soderbergh]] • [[Steven Spielberg]] • [[Steven Weinberg]] • [[Steven Wright]] • [[Stevie Nicks]] • [[Stevie Ray Vaughan]] • [[Stevie Wonder]] • [[Stewart Brand]] • [[Stieg Larsson]] • [[Stilicho]] • [[Stiller and Meara]] • [[Stilson Hutchins]] • [[Sting (musician)]] • [[Stirling Moss]] • [[Stobaeus]] • [[Stokely Carmichael]] • [[Stone Cold Steve Austin]] • [[Stonewall Jackson]] • [[Story Musgrave]] • [[Strabo]] • [[Straub-Huillet]] • [[Stroganov family]] • [[Strom Thurmond]] • [[Stromae]] • [[Stu Hart]] • [[Stu Ungar]] • [[Stuart Broad]] • [[Stuart Hall (cultural theorist)]] • [[Stuart Macintyre]] • [[Stuart Roosa]] • [[Stuart Scott]] • [[Studs Terkel]] • [[Stéphane Breitwieser]] • [[Stéphane Grappelli]] • [[Stéphane Mallarmé]] • [[Su Bai]] • [[Su Dingfang]] • [[Su Qin]] • [[Su Shi]] • [[Su Yu]] • [[Sua Sulu'ape Paulo II]] • [[Subh-i-Azal]] • [[Subhas Chandra Bose]] • [[Subrahmanyan Chandrasekhar]] • [[Subutai]] • [[Sudas]] • [[Sudhir Kumar Chaudhary]] • [[Sudirman]] • [[Sudirman (singer)]] • [[Sue Grafton]] • [[Suetonius]] • [[Sugar Ray Leonard]] • [[Sugar Ray Robinson]] • [[Suge Knight]] • [[Suger]] • [[Suharto]] • [[Suhasini Maniratnam]] • [[Suhungmung]] • [[Sujatha (actress)]] • [[Sukarno]] • [[Suleiman the Magnificent]] • [[Sulla]] • [[Sully Prudhomme]] • [[Sulpicia (satirist)]] • [[Sultan Agung of Mataram]] • [[Sultan bin Muhammad Al-Qasimi]] • [[Sultan bin Salman Al Saud]] • [[Sumitra Peries]] • [[Sun Bin]] • [[Sun Li-jen]] • [[Sun Myung Moon]] • [[Sun Quan]] • [[Sun Ra]] • [[Sun Tzu]] • [[Sun Wen (footballer)]] • [[Sun Yat-sen]] • [[Sun Yu (director)]] • [[Sundance Kid]] • [[Sundiata Keita]] • [[Sunil Dutt]] • [[Sunil Gavaskar]] • [[Sunny Deol]] • [[Sunny Leone]] • [[Sunthorn Phu]] • [[Super Junior]] • [[Supriya Pathak]] • [[Suraiya]] • [[Suraphol Sombatcharoen]] • [[Surdas]] • [[Surena]] • [[Surendranath Dasgupta]] • [[Suroosh Alvi]] • [[Suryavarman II]] • [[Susan B. Anthony]] • [[Susan Blackmore]] • [[Susan Brownmiller]] • [[Susan Hayward]] • [[Susan Howe]] • [[Susan Lucci]] • [[Susan Sarandon]] • [[Susan Sontag]] • [[Susan Wojcicki]] • [[Susana Giménez]] • [[Susanna Centlivre]] • [[Susanna Kaysen]] • [[Susenyos I]] • [[Sushruta]] • [[Susilo Bambang Yudhoyono]] • [[Suzanne Farrell]] • [[Suzanne Lenglen]] • [[Suzy Parker]] • [[Svante Arrhenius]] • [[Svatopluk I of Moravia]] • [[Sveinn Björnsson]] • [[Sven Hedin]] • [[Sven Kramer]] • [[Sven Nykvist]] • [[Sverre Fehn]] • [[Svetlana Alexievich]] • [[Svetlana Savitskaya]] • [[Svetlana Sorokina]] • [[Svetozar Boroević]] • [[Sviatopolk II of Kiev]] • [[Sviatoslav I]] • [[Sviatoslav Richter]] • [[Swami Vivekananda]] • [[Swaminarayan]] • [[Swathi Thirunal Rama Varma]] • [[Sweyn Forkbeard]] • [[Syagrius]] • [[Sydney Barnes]] • [[Sydney Omarr]] • [[Sydney Pollack]] • [[Sydney Possuelo]] • [[Syed Ahmad Khan]] • [[Syl Cheney-Coker]] • [[Sylvanus Olympio]] • [[Sylvester Graham]] • [[Sylvester H. Roper]] • [[Sylvester Stallone]] • [[Sylvia Browne]] • [[Sylvia Chang]] • [[Sylvia Plath]] • [[Sylvia Rivera]] • [[Sylvia Robinson]] • [[Sylvie Guillem]] • [[Sylvie Vartan]] • [[Symon Petliura]] • [[Syngman Rhee]] • [[Sándor Kocsis]] • [[Sándor Kőrösi Csoma]] • [[Sándor Petőfi]] • [[Sébastien Chabal]] • [[Sébastien Le Prestre de Vauban]] • [[Sébastien Loeb]] • [[Sócrates]] • [[Sônia Braga]] • [[Søren Kierkegaard]] • [[Süleyman Demirel]] • [[Sālote Tupou III]] • [[Sławomir Mrożek]] • [[Sơn Tùng M-TP]]
3333 Munteanu had his first film role in ''Bogat'', a 2016 German-Romanian film shot in Munich.&lt;ref name=facts/&gt; His breakthrough in acting came in 2018, when [[Sylvester Stallone]] was looking for a European heavyweight boxer to play Viktor Drago, the son of [[Ivan Drago]], in the sports drama sequel ''Creed II''. Stallone found Munteanu through training videos on the Internet and personally promoted him for the role.&lt;ref name=r1/&gt; At that time Munteanu was {{convert|6|ft|4|in|cm|abbr=on}} tall and weighed ca. {{convert|245|lb|kg|abbr=on}}, but had to shed ca. {{convert|20|lb|kg|abbr=on}} for the role.&lt;ref name=r3/&gt;
3334 ==== {{lat|Sylvester Stallone}} ====
3335 ==== {{lat|Sylvester Stallone}} ====
3336 * {{icon|B}} [[:Sylvester Stallone]]
3337 |text=Tom Fitzgerald and I are not brothers. We look nothing alike. I'm considered a combination of [[John Wayne]], [[Sylvester Stallone]] and [[Paul Newman]], whereas Tom is... well... pretty much your average, forgettable type... I bring up the brother business merely to forestall the occasional cynic who might say I wrote this piece simply because I'm related to the subject. True, I call him Uncle Tom, but only out of respect, as [[Amy Carter]] would call [[Hamilton Jordan]] Uncle Ham... His story-telling talent made Fitzgerald a frequent between-periods radio and television guest, and he reportedly holds the [[Guinness World Records|Guinness record]] for the accumulation of complimentary raincoats, Panasonic radios and [[A. T. Cross Company|Cross]] pens.
3338 After a long spell performing in direct-to-video films since 1995, Lundgren returned to Hollywood in 2010 with the role of [[List of The Expendables characters#Gunner Jensen|Gunnar Jensen]] in ''[[The Expendables (2010 film)|The Expendables]]'', alongside [[Sylvester Stallone]] and an all-action star cast. He reprised his role in ''[[The Expendables 2]]'' (2012) and ''[[The Expendables 3]]'' (2014). Also in 2014, he co-starred in ''[[Skin Trade (film)|Skin Trade]]'', an action thriller about human trafficking he co-wrote and produced. He reprised his role of Ivan Drago in ''[[Creed II]]'' (2018), and is due to reprise his role as Gunner Jensen in ''[[Expend4bles]]''.&lt;ref name=&quot;Expendables4Lund&quot;&gt;Wales, George. [https://www.gamesradar.com/expendables-4-will-arrive-2017/ &quot;Confirmed: ''The Expendables 4'' will arrive in 2017&quot;], ''www.gamesradar.com'', published 10 December 2015. Retrieved 15 October 2015.&lt;/ref&gt;&lt;ref name=&quot;GuerraVic&quot;&gt;Guerra, Victoria. [http://www.foodworldnews.com/articles/43393/20151012/the-expendables-2015-4th-installment-shooting-next-year-which-mercenaries-return-for-sequel.htm &quot;''The Expendables'' 2015: 4th Installment Shooting Next Year, Which Mercenaries Return For Sequel?&quot;], ''www.foodworldnews.com'', published 10 December 2015. Retrieved 15 October 2015.&lt;/ref&gt; He appears in ''[[Sharknado 5: Global Swarming]]'' (2017), playing the protagonist's son Gil as an adult, and in ''[[Aquaman (film)|Aquaman]]'' (2018), playing the father of [[Mera (comics)|Mera]]. He also had a recurring role in the [[Arrow season 5|fifth season]] of ''[[Arrow (TV series)|Arrow]]''.
3339 The first date takes place three years after the film's release, which places the film entirely in the near future. The date appears on a computer screen as the day when [[Sylvester Stallone]]'s character John Spartan is cryogenically imprisoned for crimes he did not commit.
3340 Winkler then moved back to New York, working in theater, television commercials, and the independent film ''[[The Lords of Flatbush]]'' (1974), with then unknown [[Sylvester Stallone]]. After a short period in Los Angeles, he won the role of [[Fonzie|Arthur &quot;Fonzie&quot; Fonzarelli]] (on the 1974-1984 American television series ''[[Happy Days]]''). He also portrayed Jack Dunne in ''[[Heroes (1977 film)|Heroes]]'' (1977), Andy in the [[Carl Reiner]] film ''[[The One and Only (1978 film)|The One and Only]]'' (1978), and Chuck Lumley in [[Ron Howard]]'s ''[[Night Shift (1982 film)|Night Shift]]'' (1982). In addition, he gained recognition as an executive producer for works such as ''[[MacGyver (1985 TV series)|MacGyver]],'' ''[[Dead Man's Gun]]'', the ''[[CBS Schoolbreak Special#Season 1 (January 24 - June 12, 1984)|CBS Schoolbreak Special: &quot;All the Kids Do It&quot;]],'' ''[[Hollywood Squares]],'' and ''[[Who Are the DeBolts? And Where Did They Get Nineteen Kids?]]''
3341 |Early role for [[Sylvester Stallone]]
3342 |title=Henry Winkler on how Sylvester Stallone inspired the Fonz
3343 '''Stakar Ogord''' (portrayed by [[Sylvester Stallone]]) is a legendary Ravager captain and the leader of the Stakar Ravager Clan. Ogord saved Yondu Udonta from a life of slavery to the Kree and welcomed him to the Ravagers, but was later forced to exile him for engaging in child trafficking, thus violating the Ravager code. He joins the other Ravager clans at honoring Yondu during his funeral after his sacrifice to save Peter Quill. In a mid-credits scene, Stakar and Martinex reunite with their old teammates Charlie-27, Aleta Ogord, Mainframe, and Krugarr. He later aids the Guardians in their infiltration of the Orgoscope space station during their search of a passkey to disable the kill switch in Rocket's heart and save his life.
3344 |  ''[[Run Sweetheart Run]]'' || [[Shana Feste]] || [[Ella Balinska]], [[Pilou Asbæk]], [[Clark Gregg]], [[Aml Ameen]], [[Dayo Okeniyi]], [[Betsy Brandt]], [[Shohreh Aghdashloo]] || United States || Thriller&lt;ref&gt;{{cite web|first=Anthony|last=D'Alessandro|title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan', 'Good Night Oppy' &amp; More|url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/|date=August 1, 2022|website=Deadline Hollywood|access-date=August 3, 2022}}&lt;/ref&gt;
3345 | ''[[Samaritan (2022 film)|Samaritan]]'' || [[Julius Avery]] || [[Sylvester Stallone]], [[Martin Starr]], [[Moisés Arias]], [[Dascha Polanco]] || United States ||Superhero thriller&lt;ref&gt;{{cite web|first=Jon|last=Fuge|title=Sylvester Stallone Superhero Flick Samaritan Sets 2022 Release Date|url=https://movieweb.com/samaritan-release-date-sylvester-stallone/|date=August 11, 2021|website=Movie Web|access-date=October 21, 2021}}&lt;/ref&gt;
```



```
3346 ¦ ''[[Expend4bles]]'' ¦¦ [[Scott Waugh]] ¦¦ [[Sylvester Stallone]], [[Jason Statham]], [[Dolph Lundgren]], [[Randy Couture]], [[Megan Fox]] [[50 Cent]], [[Andy Garcia]], [[Tony Jaa]], [[Eddie Hall]], Sheila Shah, [[Jacob Scipio]], [[Levy Tran]], [[Iko Uwais]] ¦¦ United States ¦¦Action thriller&lt;ref&gt;{{cite web¦url=https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video¦title=Expendables 4: Sylvester Stallone Looks Thoroughly Badass In New Set Video¦first=Erik¦last=Swann¦date=October 12, 2021¦work=Cinema Blend¦access-date=October 26, 2021}}&lt;/ref&gt;
3347 ¦ ''[[Levon's Trade]]'' ¦¦ [[David Ayer]] ¦¦ [[Jason Statham]], [[David Harbour]], [[Michael Peña]], [[Jason Flemyng]], Arianna Rivas ¦¦ United States ¦¦Action thriller&lt;ref&gt;{{cite web¦last=Wiseman¦first=Andreas¦title=Michael Peña, David Harbour, Jason Flemyng &amp; Arianna Rivas Join Jason Statham In David Ayer's 'Levon's Trade'; Amazon MGM Sets Release Date For Black Bear Action Pic Scripted By Sylvester Stallone¦url=https://deadline.com/2024/04/david-ayer-jason-statham-slyvester-stallone-movie-levons-trade-begins-amazon-mgm-release-date-1235878520/¦date=April 8, 2024¦website=Deadline Hollywood¦access-date=May 28, 2024}}&lt;/ref&gt;
3348 ¦ ''[[Alarum (film)¦Alarum]]'' ¦¦ [[Polish brothers¦Michael Polish]] ¦¦ [[Sylvester Stallone]], [[Scott Eastwood]], [[Willa Fitzgerald]], [[Mike Colter]], [[Ísis Valverde]], [[D. W. Moffett]] ¦¦ United States ¦¦Action crime thriller&lt;ref&gt;{{cite web¦first=Alex¦last=Ritman¦title=Sylvester Stallone Action Thriller 'Alarum' Acquired by Signature for U.K. (EXCLUSIVE)¦url=https://variety.com/2024/film/global/sylvester-stallones-alarum-signature-uk-1236005945/¦date=May 17, 2024¦work=Variety¦access-date=June 6, 2024}}&lt;/ref&gt;
3349 ¦ ''[[Armored (upcoming film)¦Armored]]'' ¦¦ Justin Routt ¦¦ [[Sylvester Stallone]], [[Jason Patric]], [[Josh Wiggins]], [[Dash Mihok]] ¦¦ United States ¦¦Action thriller&lt;ref&gt;{{cite web¦author1=Meg James¦author2=Amy Kaufman¦title=After scandal, movie producer Randall Emmett is flying under the radar with a new name¦url=https://www.latimes.com/entertainment-arts/business/story/2024-04-26/randall-emmett-ives-sylvester-stallone¦date=April 26, 2024¦website=[[LA Times]]¦access-date=June 6, 2024}}&lt;/ref&gt;
3350 *[[:Sylvester Stallone]] (1 edit): [[Special:Diff/848408173¦(+950)]]
3351 ¦ [[Sylvester Stallone]]&lt;br&gt;''[[Creed (film)¦Creed]]''
3352 ¦ co-production with [[Millennium Media]], Templeton Media, [[Campbell Grobman Films¦Grobman Films]] and Media Capital Technologies&lt;ref&gt;{{cite web¦last1=D'Alessandro¦first1=Anthony¦title='The Expendables' Returning For Lionsgate With Sylvester Stallone, Jason Statham, Dolph Lundgren, Megan Fox &amp; More¦url=https://deadline.com/2021/08/the-expendables-returning-for-lionsgate-with-sylvester-stallone-jason-statham-dolph-lundgren-megan-fox-more-1234824142/¦website=[[Deadline Hollywood]]¦access-date=August 30, 2021¦date=August 30, 2021¦archive-date=December 13, 2021¦archive-url=https://web.archive.org/web/20211213191205/https://deadline.com/2021/08/the-expendables-returning-for-lionsgate-with-sylvester-stallone-jason-statham-dolph-lundgren-megan-fox-more-1234824142/¦url-status=live}}&lt;/ref&gt;
3353 ¦&lt;ref&gt;{{cite web ¦last=Pedersen ¦first=Erik ¦title=Sylvester Stallone's 'Tulsa King' Gets Paramount+ Premiere Date, Simultaneous Cable Launch¦url=https://deadline.com/2022/05/sylvester-stallone-tulsa-king-paramount-premiere-date-paramount-plus-yellowstone-1235027354/ ¦website=[[Deadline Hollywood]] ¦date=May 18, 2022 ¦access-date=May 18, 2022}}&lt;/ref&gt;
3354 Netflix Offloads Two Completed Films, Filmmakers Shop Projects Elsewhere¦publisher=The Hollywood Reporter¦date=January 27, 2023¦access-date=August 3, 2024¦archive-date=January 28, 2023¦archive-url=https://web.archive.org/web/20230128003549/https://www.hollywoodreporter.com/movies/movie-news/netflix-offloads-inheritance-house-wife-1235310514/¦url-status=live}}&lt;/ref&gt; In August 2021, Cohen gained notability from writing the screenplay and story for ''[[Expend4bles]]''.&lt;ref name=&quot;TE4&quot;&gt;{{cite web¦url=https://www.hollywoodreporter.com/movies/movie-news/jason-statham-sylvester-stallone-50-cent-megan-fox-new-expendables-movie-cast-1235004992/¦work=[[The Hollywood Reporter]]¦title=Jason Statham, Sylvester Stallone, 50 Cent, Megan Fox to Star in New 'The Expendables' Movie for Lionsgate (Exclusive)¦first=Borys ¦last=Kit¦date=August 30, 2021¦access-date=December 12, 2021}}&lt;/ref&gt; In March 2022, he wrote and directed the horror short film ''[[Blink (2022 film)¦Blink]]''.&lt;ref name=&quot;Blink&quot; /&gt; In May 2024, he made his feature-length directorial debut with the horror film ''[[Tarot (2024 American film)¦Tarot]]'' (originally ''Horrorscope''), which he also wrote.&lt;ref name=&quot;:2&quot; /&gt;&lt;ref&gt;{{cite web¦url=https://www.varietyinsight.com/production_restart_full_view.php?track_id=179925 ¦title=Film and TV Projects Going Into Production - Horrorscope ¦website=[[Variety Insight]] ¦access-date=December 12, 2021¦url-status=live¦archive-url=https://web.archive.org/web/20210516200338/https://www.varietyinsight.com/production_restart_full_view.php?track_id=179925 ¦archive-date=May 16, 2021}}&lt;/ref&gt;
3355 ¦ &lt;ref name=&quot;TE4&quot; /&gt;&lt;ref&gt;{{cite web¦url=https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video¦title=Expendables 4: Sylvester Stallone Looks Thoroughly Badass In New Set Video¦first=Erik¦last=Swann¦work=[[CinemaBlend]]¦date=October 12, 2021¦access-date=December 12, 2021¦url-status=live¦archive-date=October 14, 2021¦archive-url=https://web.archive.org/web/20211014065910/https://www.cinemablend.com/movies/expendables-4-sylvester-stallone-looks-thoroughly-badass-in-new-set-video}}&lt;/ref&gt;
3356 ¦ The [[Justice League]] wasn't the only DCEU property that saw redemption this year. In 2016, [[David Ayer]] made ''Suicide Squad'', and it was terrible. Jump to 2021, when [[James Gunn]] (director of Marvel's ''[[Guardians of the Galaxy (film)¦Guardians of the Galaxy]]'' who jumped ship to DC after Disney temporarily fired him for some unsavory old tweets) made '''''The''' Suicide Squad'', a sequel/reboot/some weird combination of the two... and it was amazing! ''The Suicide Squad'' retains some [[Suicide Squad]] members from ''Suicide Squad'' (of course, Warner Bros. would never get rid of [[Margot Robbie]]'s [[Harley Quinn]]) but mostly follows a new set of characters including [[Idris Elba]]'s [[Bloodsport (character)¦Bloodsport]], [[Sylvester Stallone]]'s [[King Shark]], and [[Daniela Melchior]]'s [[Ratcatcher (comics)¦Ratcatcher]] as they attempt to destroy [[Starro the Conquerer]], a giant alien starfish.
3357 *****I don't see how you inferred that, but no. If Clint Eastwood or Sylvester Stallone dies tomorrow with nothing to blurb but job description and age, it'd be the same. Stories need hooks. RD is for simple recent deaths in the news. It has nothing to do with life beforehand. That part just determines article creation and content here. [[User:InedibleHulk¦InedibleHulk]] ([[User talk:InedibleHulk¦talk]]) 15:12, 23 January 2022 (UTC)
3358 ¦Guests=[[Sylvester Stallone]], [[Linda Cardellini]]
3359 ¦ShortSummary=Audience is made up of veterans; Medals for Regular People; [[Snoop Dogg¦Snoop]]'s Movie Titles; Thank You Notes; Freestylin' with The Roots; Sylvester Stallone teaches Jimmy how to fake a punch; Louis Tomlinson performed &quot;Silver Tongues&quot;
3360 ** [[Sylvester Stallone]] - ''[[The Suicide Squad (film)¦The Suicide Squad]]'' as [[King Shark¦Nanaue / King Shark]]
3361 * '''''[[Tulsa King]]'''''{{efn¦U.S. broadcast television premiere; a [[Paramount+]] original series.}}&lt;ref&gt;{{Cite press release¦title=Season One of Paramount+ Original Series &quot;Tulsa King,&quot; Starring Oscar Nominee Sylvester Stallone, to Make Broadcast Premiere on CBS in Summer 2024¦url=http://www.thefutoncritic.com/news/2024/02/28/season-one-of-paramountplus-original-series-tulsa-king-starring-oscar-nominee-sylvester-stallone-to-make-broadcast-premiere-on-cbs-in-summer-2024-262210/20240228cbs01/¦publisher=[[CBS]]¦via=[[The Futon Critic]]¦date=February 28, 2024}}&lt;/ref&gt;
3362 In June 2021, it was announced that Idoko was cast in the supporting role of Sallay in [[Blumhouse Productions¦Blumhouse]] and [[Amazon Studios¦Amazon Studios']] psychological Horror film, ''Nanny'',&lt;ref&gt;{{Cite web¦last=N'Duka¦first=Amanda¦date=2021-04-13¦title=Stay Gold Features &amp; Topic Studios Team On Horror Film 'Nanny' From Nikyatu Jusu¦url=https://deadline.com/2021/04/stay-gold-features-topic-studios-nanny-nikyatu-jusu-1234733360/¦access-date=2022-01-25¦website=Deadline¦language=en-US}}&lt;/ref&gt; Which Premiered at [[Sundance Film Festival]]&lt;ref&gt;{{Cite magazine¦date=2022-01-20¦title=The 10 Films We're Most Excited to See at Sundance¦url=https://www.vanityfair.com/hollywood/2022/01/the-10-films-were-most-excited-to-see-at-sundance¦access-date=2022-01-25¦magazine=Vanity Fair¦language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite news¦last=Sperling¦first=Nicole¦date=2021-12-09¦title=Sundance Film Festival Unveils 2022 Lineup That Reflects 'Age of Reckoning'¦language=en-US¦work=The New York Times¦url=https://www.nytimes.com/2021/12/09/movies/sundance-2022-lineup.html¦access-date=2022-01-25¦issn=0362-4331}}&lt;/ref&gt; 2022 and won the [[Sundance Grand Jury Prize¦Grand Jury Prize]]&lt;ref&gt;{{Cite web¦last=Galuppo¦first=Mia¦date=2022-01-29¦title=Sundance: 'Nanny' and 'The Exiles' Take Top Prizes at 2022 Fest¦url=https://www.hollywoodreporter.com/movies/movie-news/sundance-film-festival-winners-2022-nanny-exiles-1235083659/¦access-date=2022-01-30¦website=The Hollywood Reporter¦language=en-US}}&lt;/ref&gt; in the dramatic competition. Idoko has discussed the film and her career in a number of interviews, including her feature in [https://www.photobookmagazine.com/features/zephani-idoko Photobook Magazine].&lt;ref&gt;{{Cite web ¦last=Seth ¦first=Kenisha ¦date=2021 ¦title=Zephani Idoko stars as Abena in &quot;The Unsettling&quot; - a psychological horror film ¦url=https://www.photobookmagazine.com/features/zephani-idoko ¦access-date=2022-06-23 ¦website=PhotoBook Magazine ¦language=en-US}}&lt;/ref&gt; and with [[OkayAfrica]]&lt;ref&gt;{{Cite web ¦title=Zephani Idoko Talks 'Nanny,' 'Gossip Girl,' and Making Horror Films - OkayAfrica ¦url=https://www.okayafrica.com/nanny-film-zephani-idoko/ ¦access-date=2023-01-04 ¦website=www.okayafrica.com ¦language=en}}&lt;/ref&gt; Nanny screened in the Special Presentations category at [[Toronto International Film Festival¦TIFF]] in September 2022&lt;ref&gt;{{Cite web ¦last=D'Alessandro ¦first=Anthony ¦date=2022-07-28 ¦title=Toronto Film Festival Expecting &quot;Full Houses&quot;, Says CEO With Star Power World Premieres: Jennifer Lawrence's 'Causeway', Anya Taylor-Joy's 'The Menu', Jessica Chastain's 'The Good Nurse', Jonathan Majors &amp; Glen Powell's 'Devotion' &amp; More ¦url=https://deadline.com/2022/07/2022-toronto-film-festival-lineup-movies-list-1235079516/ ¦access-date=2022-08-09 ¦website=Deadline ¦language=en-US}}&lt;/ref&gt; and at several other festivals including [[BFI London Film Festival]],&lt;ref&gt;{{Cite web ¦last=Stottor ¦first=William ¦date=2022-10-08 ¦title=Nanny: Interview with Anna Diop, Michelle Monaghan &amp; Zephani Idoko ¦url=https://loudandclearreviews.com/nanny-interview-anna-diop-michelle-monaghan-zephani-idoko-2022-movie-red-carpet-film/ ¦access-date=2023-01-04 ¦website=Loud And Clear Reviews ¦language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦last=Bradbury ¦first=Sarah ¦title=Nanny: On the red carpet with Anna Diop, Michelle Monaghan, Nikyatu Jusu and Zephani Idoko at London Film Festival 2022 ¦url=https://www.theupcoming.co.uk/2022/10/15/nanny-on-the-red-carpet-with-anna-diop-michelle-monaghan-nikyatu-jusu-and-zephani-idoko-at-london-film-festival-2022/ ¦access-date=2023-01-04 ¦website=The Upcoming}}&lt;/ref&gt; [[AFI Fest¦AFI FEST]]&lt;ref&gt;{{Cite web ¦title=AFI FEST 2022 {{!}} NANNY ¦url=https://fest.afi.com/movies/nanny/ ¦access-date=2023-01-04 ¦website=AFI FEST ¦language=en}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦title=AFI FEST 2022 {{!}} Photos ¦url=https://fest.afi.com/photos/ ¦access-date=2023-01-04 ¦website=AFI FEST ¦language=en}}&lt;/ref&gt; and [[Africa International Film Festival¦AFRIFF]]&lt;ref&gt;{{Cite web ¦title=Opening Films ¦url=https://www.afriff.com/openingfilms ¦access-date=2023-01-04 ¦website=Afriff ¦language=en}}&lt;/ref&gt; in Lagos Nigeria, where Idoko was born. Nanny was released in movie theaters November 23, 2022,&lt;ref&gt;{{Cite web ¦last=D'Alessandro ¦first=Anthony ¦date=2022-08-01 ¦title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan', 'Good Night Oppy' &amp; More ¦url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/ ¦access-date=2022-08-09 ¦website=Deadline ¦language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Citation ¦title=Nanny ¦date=2022-11-22 ¦url=http://www.amctheatres.com/movies/nanny-70617 ¦language=en ¦access-date=2022-09-30}}&lt;/ref&gt; followed by a [[Prime Video]] streaming release on December 16, 2022.&lt;ref&gt;{{Cite web ¦title=Nanny ¦url=https://press.amazonstudios.com/us/en/original-movies/nanny ¦access-date=2022-08-09 ¦website=press.amazonstudios.com}}&lt;/ref&gt;
3363 American actor, film producer, and [[martial arts¦martial artist]] [[Wesley Snipes]] started his acting career in the 1980s with a one episode appearance on the [[soap opera]] ''[[All My Children]]'' (1984). His first film roles were both in 1986 in ''[[Wildcats (film)¦Wildcats]]'' and ''[[Streets of Gold (film)¦Streets of Gold]]''. He then appeared as Willie Mays Hayes in the 1989 [[Sports film¦sports]] [[comedy film]] ''[[Major League (film)¦Major League]]'' with [[Tom Berenger]] and [[Charlie Sheen]]. He had prominent roles in the 1990s, starring in the films ''[[New Jack City]]'' with [[Ice-T]] (1991), ''[[Jungle Fever]]'' with [[Annabella Sciorra]] and [[Anthony Quinn]] (1991), ''[[White Men Can't Jump]]'' with [[Woody Harrelson]] (1992), ''[[Rising Sun (film)¦Rising Sun]]'' with [[Sean Connery]] (1993), ''[[Demolition Man (film)¦Demolition Man]]'' with [[Sylvester Stallone]] (1993), ''[[To Wong Foo, Thanks for Everything! Julie Newmar]]'' with [[Patrick Swayze]] and [[John Leguizamo]] (1995), and ''[[U.S. Marshals (film)¦U.S. Marshals]]'' with [[Tommy Lee Jones]] (1998). Also in 1998, he was cast as [[Blade (New Line film series)¦Eric Brooks / Blade]] in the film ''[[Blade (1998 film)¦Blade]]'', based on the [[Marvel Comics]] [[Blade (character)¦superhero of the same name]], a role he went on to reprise in ''[[Blade II]]'' (2002), ''[[Blade: Trinity]]'' (2004) and ''[[Deadpool &amp; Wolverine]]'' (2024).
3364 Snipes had smaller roles during the 2000s, most of them being [[direct-to-video]] before returning to the theatrical release, cast as [[List of The Expendables characters#Doctor &quot;Doc&quot; Death¦Doctor &quot;Doc&quot; Death]] in the 2014 [[action film]] ''[[The Expendables 3]]'' where he was reunited working with Sylvester Stallone. He teamed up with [[Eddie Murphy]] to co-star in the
```



[[biographical film|biographical]] [[comedy film]] ''[[Dolemite Is My Name]]'' as [[D'Urville Martin]] (2019) and the [[comedy film]] ''[[Coming 2 America]]'' (2021), as General Izzi.

3365 The film marks First Films' first venture into local production after distribution of Hong Kong-based action movies to the Philippines.&lt;ref&gt;{{cite web|url=https://news.google.com/newspapers?id=i4YqAAAAIBAJ&amp;sjid=BAsEAAAAIBAJ&amp;pg=6503%2C521928|last=Balsas|first=Zar|title=First Films' Multi-Million Peso Production of &quot;Dalawa Man ang Buhay Mo, Pagsasabayin Ko&quot; is at Par with Any Movie of Sylvester Stallone, Arnold Schwarzenegger, Cindy Rothrock and Jean-Claude Van Damme|pages=17|date=July 4, 1991|newspaper=[[Manila Standard]]|publisher=Philippine Manila Standard Publishing|accessdate=February 26, 2022|via=Google News}}&lt;/ref&gt;

3366 Muscedere was born in [[Windsor, Ontario]] in 1959, the son of [[Italian Canadians|Italian immigrants]], moving to [[Chatham, Ontario|Chatham]] at the age of 12.{{sfn|Edwards|2010|pp=65 &amp; 67}} His parents, Domenico and Ortensia Muscedere, were from Vicalvi-Pozzuoli, [[Frosinone]] near [[Lazio]], and left Italy in 1957 to escape post-war poverty.{{sfn|Edwards|2010|p=65}} Muscedere grew up in an Italian-speaking household and he always spoke with an [[Italian accent]], which he tried to hide by mumbling.{{sfn|Edwards|2010|p=66}} Those who knew him described him as sounding like [[Sylvester Stallone]]'s character [[Rocky Balboa]].{{sfn|Edwards|2010|p=66}} As a child, he was often bullied for being a &quot;[[wop]]&quot; and a &quot;[[List of ethnic slurs#D|dogan]]&quot; (derogatory Canadian slang for a Roman Catholic), and fought back, acquiring a great reputation as a fighter, which led him to take up boxing as his hobby.{{sfn|Edwards|2010|pp=66-67}} Muscedere's friends and family believed the childhood bullying had scarred him, making him adopt a tough persona to hide any weaknesses, which explained his love of boxing and later motorcycles.{{sfn|Edwards|2010|pp=66-67}} One who knew him stated: &quot;He didn't really want to be somebody anybody could look down on. He wanted respect&quot;.{{sfn|Edwards|2010|p=66}} Muscedere enjoyed some success as an amateur boxer, but his early marriage and his first child's birth prevented him from becoming a professional boxer as he would have liked, forcing him to take a job in order to support his family.{{sfn|Edwards|2010|p=67}} Muscedere worked at a factory making automobile brakes in Chatham from 1979 until his murder, a job he hated as he found it very dull, but it gave him a good salary.{{sfn|Edwards|2010|p=68}} After his first and second wives divorced him and were awarded custody of their children, Muscedere took up riding motorcycles, finding in the world of outlaw biking a surrogate family for those he had lost with his divorces.{{sfn|Edwards|2010|pp=68-69}} Muscedere had a strong sense of masculinity informed by traditional Italian values, for instance always paying the child support payments to both his ex-wives on time and in full on the grounds it was dishonourable for a father to let down his children.{{sfn|Edwards|2010|p=68}} In 1997, Muscedere joined the [[Annihilators Motorcycle Club]] led by [[Wayne Kellestine|Wayne &quot;Weiner&quot; Kellestine]].{{sfn|Edwards|2010|p=69}} At the time, Muscedere was in a stressful second divorce and he came to find that Kellestine's farmhouse at 32196 Aberdeen Line outside of [[Iona Station]] to be a refuge from his troubles.{{sfn|Edwards|2010|p=69-70}}

3367 {{cite book |first=Christopher |last=Pinney |title='Photos of the Gods': The Printed Image and Political Struggle in India |publisher=Reaktion Books |year=2004|isbn=978-1-86189-184-6 |url=https://books.google.com/books?id=8hhXq7hpzSwC&amp;pg=PA205 |pages=204-205 |quote=One of the most striking examples ... is Anuradha Kapur's celebrated analysis of the 'muscularization' of the god Ram. Starting in the late 1980s a series of popular images (illus. 161) appeared depicting a saffron-clad Ram (often towering above a new imaginary Rama temple in Ayodhya). Ram was endowed with a muscular armature to rival that of his simian assistant, Hanuman, and Sylvester Stallone... The angry Rama image, for instance, was first produced by the Vishva Hindu Parishad in the late 1980s and, ... several other companies produced similar images.}}&lt;/ref&gt;&lt;ref&gt;{{cite book |first1=Kajri |last1=Jain |title=Gods in the Bazaar: The Economies of Indian Calendar Art |publisher=Duke University Press |year=2007 |isbn=978-0-8223-3926-7 |url=https://books.google.com/books?id=B7LEXbIREjEC |pages=320-321 |quote=[Anuradha] Kapur traces a marked iconographic shift in popular imagery from the earlier, textually sanctioned depictions of Ram as soft, smooth-bodied, almost pudgy, smiling, benign, and above all gentle and tranquil (see figs. 93 and 129), to the more recent muscular versions whose ''rasa'' or mood is (according to Kapur) predominantly ''ugra'': &quot;angry, exercised ... punishing&quot; (75), emphasizing his bow and arrows in their capacity as weapons rather than as mere iconographic markers (fig. 138).... The departure from these established iconographic conventions, Kapur argues, is made possible by &quot;the making of a virile Hinduism,&quot; which accompanies the encroachment of &quot;realism,&quot; and particularly the depiction of a &quot;virile&quot; physiognomy, onto the mythic or iconic image.}}&lt;/ref&gt;]]

3368 Stuart was hired by [[20th Century Fox]] to adapt Thorp's novel into a screenplay in 1987. His finished draft was [[greenlit]] immediately by Fox, which was eager for a [[Blockbuster (entertainment)|summer blockbuster]] the following year. The role of McClane was turned down by a host of the decade's most popular actors, including [[Arnold Schwarzenegger]]&lt;nowiki/&gt;and [[Sylvester Stallone]]. Known mainly for work on television, Willis was paid $5 million for his involvement, placing him among Hollywood's highest-paid actors. The deal was seen as a poor investment by industry professionals and attracted significant controversy before its release. Filming took place between November 1987 and March 1988, on a $25 million-$35 million budget and almost entirely on location in and around [[Fox Plaza (Los Angeles)|Fox Plaza]] in Los Angeles.

3369 [[Emmett/Furla Oasis]] (EFO) is considered a prolific producer of these films; [[Bruce Willis]] starred in several EFO films from 2011 (''[[Setup (2011 film)|Setup]]'') until his 2022 retirement from acting, typically only working for one or two filming days per film, and sometimes appearing for as little as seven minutes of screen time as in the 2020 film ''[[Hard Kill]]''.&lt;ref name=&quot;vul-emmett&quot;&gt;{{cite web|url=https://www.vulture.com/article/randall-emmett-movies.html|title=The King of the Geezer Teasers|first=Joshua|last=Hunt|work=[[Vulture (website)|Vulture]]|date=March 31, 2021|access-date=March 30, 2022}}&lt;/ref&gt;&lt;ref&gt;{{cite web |last1=Albert |first1=Victoria |title=Bruce Willis is &quot;stepping away&quot; from his acting career due to aphasia diagnosis, family says |url=https://www.cbsnews.com/news/bruce-willis-aphasia-stepping-away-acting-career/ |website=CBS News |access-date=March 30, 2022 |date=March 30, 2022}}&lt;/ref&gt;&lt;ref name=&quot;cbr-willis&quot;&gt;{{cite web|url=https://www.cbr.com/bruce-willis-geezer-teasers-explained/|title=What Are 'Geezer Teasers' - and Why Does Bruce Willis Have So Many?|first=Timothy|last=Donohoo|work=[[CBR.com]]|date=May 5, 2022|access-date=October 19, 2022}}&lt;/ref&gt; Other actors commonly associated with these films include [[Nicolas Cage]], [[Mel Gibson]], [[Steven Seagal]], [[Sylvester Stallone]], and [[John Travolta]].&lt;ref name=&quot;vul-emmett&quot;/&gt;&lt;ref name=&quot;cbr-willis&quot; /&gt; Although such movies are rarely well-received by [[film critic]]s, [[Lionsgate]], EFO's main distributor, has stated that EFO's productions have been &quot;consistently profitable&quot;.&lt;ref name=&quot;vul-emmett&quot;/&gt;

3370 Eddie was born in [[New York City]], [[New York (state)|New York]]. He graduated from [[Long Island University]], had worked as a doorman in New York City, and then moved to Los Angeles to work as a personal trainer for [[Mickey Rourke]] and [[Anthony Michael Hall]].&lt;ref&gt;{{Cite web|url=https://www.martialartsentertainment.com/eddie-perez/|title=Eddie Perez|website=martialartsentertainment.com|date=7 September 2017 |access-date=2022-03-16}}&lt;/ref&gt; He worked as a bodyguard for [[Duran Duran]], [[Sylvester Stallone]], and [[Sandra Bullock]]. Later, he studied acting with Peter Flood and stunts with [[Paul Stader]].&lt;ref&gt;{{Cite web|url=http://www.filmfatales.org/article/148|title=STUNTS ON SCREEN|website=filmfatales.org|access-date=2022-03-16}}&lt;/ref&gt; He was the second unit director and stunt coordinator in the films such as ''[[Blade (1998 film)|Blade]]'' and ''[[Shoot 'Em Up (film)|Shoot 'Em Up]]''.&lt;ref&gt;{{Cite web|url=https://www.backstage.com/magazine/article/want-stunts-need-know-130/|title=Want to Do Your Own Stunts? Here's What You Need to Know|website=backstage.com|access-date=2022-03-16}}&lt;/ref&gt;

3371 * [[Sylvester Stallone]]

3372 * Sylvester Stallone

3373 '''''Tulsa King''''' is an American [[comedy drama|comedy]] and [[crime drama]] television series created by [[Taylor Sheridan]] for the [[Streaming media|streaming]] platform [[Paramount+]]. The series stars [[Sylvester Stallone]] in his first leading role in a scripted television series. Stallone portrays Dwight &quot;The General&quot; Manfredi, a [[Mafia capo]] who has been recently released from prison in New York and is sent to [[Tulsa, Oklahoma]], where he begins to set up a criminal organization.

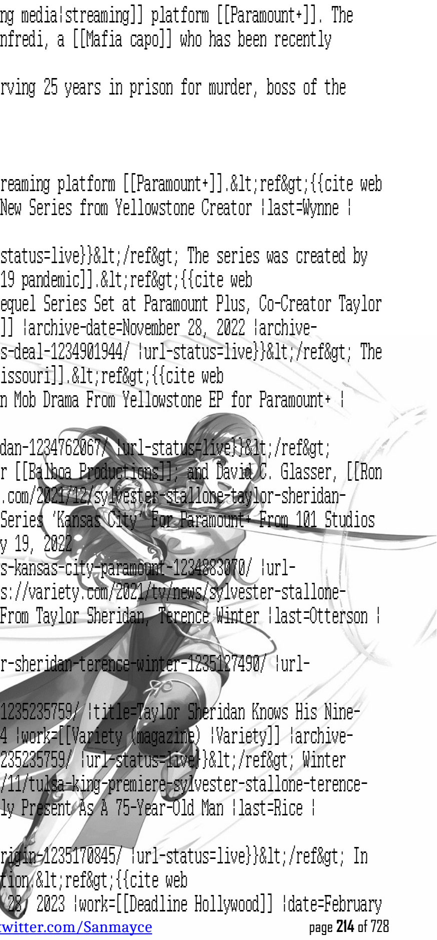
3374 * [[Sylvester Stallone]] as Dwight &quot;The General&quot; Manfredi, a [[Caporegime |capo]] in the Invernizzi family who is sent to Tulsa after serving 25 years in prison for murder, boss of the Manfredi family in Tulsa

3375 |WrittenBy = Taylor Elmore and [[Terence Winter]] &amp; [[Sylvester Stallone]]

3376 |WrittenBy = Terence Winter &amp; Sylvester Stallone

3377 On December 6, 2021, it was reported [[Taylor Sheridan]] and [[Terence Winter]] were developing a series titled ''Kansas City'' for the digital streaming platform [[Paramount+]].&lt;ref&gt;{{cite web |url=https://people.com/tv/sylvester-stallone-set-to-star-kansas-city-first-lead-tv-role/ |title=Sylvester Stallone Set for First Lead TV Role in New Series from Yellowstone Creator |last=Wynne | first=Kelly |date=December 7, 2021 |accessdate=February 28, 2024 |work=[[People (magazine) |People]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072013/https://people.com/tv/sylvester-stallone-set-to-star-kansas-city-first-lead-tv-role/ |url-status=live}}&lt;/ref&gt; The series was created by Sheridan, who had signed a multi-year contract with [[ViacomCBS]] to create new series during the [[COVID-19 pandemic in the United States |COVID-19 pandemic]].&lt;ref&gt;{{cite web |url=https://variety.com/2021/tv/news/yellowstone-prequel-series-paramount-plus-taylor-sheridan-viacomcbs-deal-1234901944/ |title='Yellowstone' Prequel Series Set at Paramount Plus, Co-Creator Taylor Sheridan Extends ViacomCBS Deal |last=Otterson |first=Joe |date=February 5, 2021 |accessdate=February 28, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=November 28, 2022 |archive-url=https://web.archive.org/web/20221128190556/https://variety.com/2021/tv/news/yellowstone-prequel-series-paramount-plus-taylor-sheridan-viacomcbs-deal-1234901944/ |url-status=live}}&lt;/ref&gt; The series would be centered around Sal, an Italian-American mobster from [[New York City]] who is tasked with returning the mafia to [[Kansas City, Missouri]].&lt;ref&gt;{{cite web |url=https://tvline.com/casting-news/sylvester-stallone-kansas-city-paramount-plus-taylor-sheridan-1234762067/ |title=Sylvester Stallone to Star in Mob Drama From Yellowstone EP for Paramount+ | last=Nemetz |first=Dave |date=December 6, 2021 |accessdate=February 28, 2024 |work=[[TVLine]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072016/https://tvline.com/casting-news/sylvester-stallone-kansas-city-paramount-plus-taylor-sheridan-1234762067/ |url-status=live}}&lt;/ref&gt; Sheridan, Winter and series-star [[Sylvester Stallone]] were announced as [[executive producer]]s alongside Braden Aftergood from Stallone's banner [[Balboa Productions]], and David C. Glasser, [[Ron Burkle]] and [[Bob Yari]] from the production company 101 Studios.&lt;ref name=&quot;EP's &amp; Stallone&quot;&gt;{{cite web |url=https://deadline.com/2021/12/sylvester-stallone-taylor-sheridan-terence-winter-drama-series-kansas-city-paramount-1234883070/ |title=Sylvester Stallone Set To Star In Taylor Sheridan &amp; Terence Winter Drama Series 'Kansas City' For Paramount+ From 101 Studios &amp; MTV Ent. Studios |last=White |first=Pater |date=December 6, 2021 |accessdate=February 28, 2024 |work=[[Deadline Hollywood]] |archive-date=May 19, 2022 |archive-url=https://web.archive.org/web/20220519015513/https://deadline.com/2021/12/sylvester-stallone-taylor-sheridan-terence-winter-drama-series-kansas-city-paramount-1234883070/ |url-status=live}}&lt;/ref&gt; Winter was also the series' [[showrunner]], which [[MTV Entertainment Studios]] produced.&lt;ref&gt;{{cite web |url=https://variety.com/2021/tv/news/sylvester-stallone-paramount-plus-series-kansas-cit-taylor-sheridan-terence-winter-1235127490/ |title=Sylvester Stallone to Star Paramount Plus Series 'Kansas City' From Taylor Sheridan, Terence Winter |last=Otterson | first=Joe |date=December 6, 2021 |accessdate=February 28, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072015/https://variety.com/2021/tv/news/sylvester-stallone-paramount-plus-series-kansas-cit-taylor-sheridan-terence-winter-1235127490/ |url-status=live}}&lt;/ref&gt;

3378 Sheridan wrote the [[pilot episode]] in a week.&lt;ref&gt;{{cite web |url=https://variety.com/2022/tv/news/taylor-sheridan-shows-yellowstone-1932-1235235759/ |title=Taylor Sheridan Knows His Nine-Show TV Slate Is 'Excessive': 'This Volume of Work Is Not Sustainable' |last=Otterson |first=Joe |date=April 20, 2022 |accessdate=February 28, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=May 9, 2022 |archive-url=https://web.archive.org/web/20220509212121/https://variety.com/2022/tv/news/taylor-sheridan-shows-yellowstone-1932-1235235759/ |url-status=live}}&lt;/ref&gt; Winter moved the show's setting to [[Tulsa, Oklahoma]], to give the characters a more-remote setting.&lt;ref&gt;{{cite web |url=https://deadline.com/2022/11/tulsa-king-premiere-sylvester-stallone-terence-winter-drama-origin-1235170845/ |title='Tulsa King' Showrunner Terence Winter Talks Mobsters, Tulsa And Sly Stallone: &quot;He Doesn't Even Remotely Present As A 75-Year-Old Man |last=Rice | first=Lynette |date=November 12, 2022 |accessdate=February 29, 2024 |work=[[Deadline Hollywood]] |archive-date=January 19, 2024 |archive-url=https://web.archive.org/web/20240119175126/https://deadline.com/2022/11/tulsa-king-premiere-sylvester-stallone-terence-winter-drama-origin-1235170845/ |url-status=live}}&lt;/ref&gt; In 2022, ''[[Deadline Hollywood]]'' stated the show was given a straight-to-series order under the title ''Tulsa King'' to reflect the change in location.&lt;ref&gt;{{cite web |url=https://deadline.com/pilot/2022-paramount-plus-series-pilot-orders/ |title=2022-23 Paramount+ Pilots &amp; Series Orders |accessdate=February 28, 2023 |work=[[Deadline Hollywood]] |date=February



15, 2022 |archive-date=May 9, 2022 |archive-url=https://web.archive.org/web/20220509212118/https://deadline.com/pilot/2022-paramount-plus-series-pilot-orders/ |url-status=live}}&lt;/ref&gt; Stallone involved himself in several aspects of the production. Winter noted: &quot;With Stallone, you're getting a writer, a director, a producer, an editor&quot;; and Glasser said Stallone was &quot;heavily involved in the editing process&quot;.&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/tv/tv-features/tulsa-king-taylor-sheridan-sylvester-stallone-1235257753/ |title=Taylor Sheridan Might Have Set a Screenwriting Record With Sylvester Stallone Show 'Tulsa King' |last=Hibberd |first=James |date=November 10, 2022 |accessdate=June 11, 2024 |work=[[The Hollywood Reporter]] |archive-date=December 16, 2023 |archive-url=https://web.archive.org/web/20231216043456/https://www.hollywoodreporter.com/tv/tv-features/tulsa-king-taylor-sheridan-sylvester-stallone-1235257753/ |url-status=live}}&lt;/ref&gt;

Winter confirmed in September 2024 that a [[spin-off (media)|spin-off]] series set in [[New Orleans]] was in the early stages of development.&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/tv/tv-features/tulsa-king-season-2-terence-winter-interview-1235990686/|title='Tulsa King' Writer Terence Winter on Season 2 Changes: “This Is Where the Fun Starts”|last=Hibberd|first=James|date=September 6, 2024|accessdate=September 29, 2024|work=[[The Hollywood Reporter]]}}&lt;/ref&gt; Later that month, Stallone confirmed that a third season of ''Tulsa King'' was also in development.&lt;ref&gt;{{cite web|url=https://screenrant.com/tulsa-king-season-3-confirmed-sylvester-stallone/|title=Sylvester Stallone Confirms Tulsa King Season 3 Is In The Works|last=Bythrow|first=Nick|date=September 25, 2024|accessdate=September 29, 2024|work=[[ScreenRant]]}}&lt;/ref&gt;

[[File:Sylvester Stallone by Gage Skidmore 2.jpg |thumb |upright=0.7 |[[Sylvester Stallone]] stars in the series lead role as Dwight &quot;The General&quot; Manfredi. |alt=Sylvester Stallone]]

At the time the series was announced, [[Sylvester Stallone]] was participating in negotiations to star as ''Tulsa King''{{'s}} lead character Sal in his scripted television debut.&lt;ref name=&quot;EP's &amp; Stallone&quot; /&gt; Stallone's character was later renamed Dwight &quot;The General&quot; Manfredi.&lt;ref name=&quot;Cast 2&quot;&gt;{{cite web |url=https://deadline.com/2022/03/tulsa-king-max-casella-domenick-lombardozzi-vincent-piazza-jay-cast-sylvester-stallone-paramount-plus-1234985490/ |title='Tulsa King': Max Casella, Domenick Lombardozzi, Vincent Piazza &amp; Jay Will Join Sylvester Stallone In Paramount+ Series |last=Andreeva |first=Nellie |date=March 24, 2022 |accessdate=February 29, 2024 |work=[[Deadline Hollywood]] |archive-date=April 15, 2022 |archive-url=https://web.archive.org/web/20220415095241/https://deadline.com/2022/03/tulsa-king-max-casella-domenick-lombardozzi-vincent-piazza-jay-cast-sylvester-stallone-paramount-plus-1234985490/ |url-status=live}}&lt;/ref&gt; Stallone stated filming television was more difficult and time-consuming than the films in which he had starred.&lt;ref&gt;{{cite web |url=https://variety.com/2022/tv/news/sylvester-stallone-tulsa-king-future-seasons-1235436078/ |title=Sylvester Stallone Isn't Sure About Future Seasons of 'Tulsa King,' Admits Making TV Is Tougher Than He Thought: Movies Are 'A Vacation' |last=Longretta |first=Emily |date=November 18, 2022 |accessdate=March 2, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=March 1, 2024 |archive-url=https://web.archive.org/web/20240301142450/https://variety.com/2022/tv/news/sylvester-stallone-tulsa-king-future-seasons-1235436078/ |url-status=live}}&lt;/ref&gt; On March 24, 2022, [[Max Casella]], [[Domenick Lombardozzi]], [[Vincent Piazza]] and [[Jay Will (actor) |Jay Will]] were added to the cast; Casella, Lombardozzi and Piazza portray members of the Invernizzi family crime syndicate while Will stars as a recent college graduate who becomes a member of Dwight's crew.&lt;ref name=&quot;Cast 2&quot; /&gt; In May, A. C. Peterson, [[Andrea Savage]], [[Garrett Hedlund]] and [[Martin Starr]] were reported to be starring in the series;&lt;ref&gt;{{cite web |url=https://variety.com/2022/tv/news/sylvester-stallone-paramount-series-tulsa-king-casts-a-c-peterson-exclusive-1235259671/ |title=Sylvester Stallone Paramount+ Series 'Tulsa King' Casts A.C. Peterson (Exclusive) |last=Otterson |first=Joe |date=May 5, 2022 |accessdate=February 29, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=May 9, 2022 |archive-url=https://web.archive.org/web/20220509232304/https://variety.com/2022/tv/news/sylvester-stallone-paramount-series-tulsa-king-casts-a-c-peterson-exclusive-1235259671/ |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://deadline.com/2022/05/tulsa-king-andrea-savage-sylvester-stallone-taylor-sheridan-1235015650/ |title='Tulsa King': Andrea Savage To Star Alongside Sylvester Stallone In Taylor Sheridan's Paramount+ Mob Drama Series |last=White |first=Peter |date=May 6, 2022 |accessdate=February 29, 2024 |work=[[Deadline Hollywood]] |archive-date=May 11, 2022 |archive-url=https://web.archive.org/web/20220511043408/https://deadline.com/2022/05/tulsa-king-andrea-savage-sylvester-stallone-taylor-sheridan-1235015650/ |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://variety.com/2022/tv/news/sylvester-stallone-paramount-plus-tulsa-king-cast-garrett-hedlund-1235264069/ |title=Sylvester Stallone Paramount+ Series 'Tulsa King' Casts Garrett Hedlund (Exclusive) |last=Otterson |first=Joe |date=May 10, 2022 |accessdate=February 29, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=July 20, 2023 |archive-url=https://web.archive.org/web/20230720144924/https://variety.com/2022/tv/news/sylvester-stallone-paramount-plus-tulsa-king-cast-garrett-hedlund-1235264069/ |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://deadline.com/2022/05/tulsa-king-martin-starr-taylor-sheridans-sylvester-stallone-led-paramount-mob-drama-series-1235020378/ |title='Tulsa King': Martin Starr Joins Taylor Sheridan's Sylvester Stallone-Led Paramount+ Mob Drama Series |last=White |first=Peter |date=May 10, 2022 |accessdate=February 29, 2024 |work=[[Deadline Hollywood]] |archive-date=March 26, 2023 |archive-url=https://web.archive.org/web/20230326223800/https://deadline.com/2022/05/tulsa-king-martin-starr-taylor-sheridans-sylvester-stallone-led-paramount-mob-drama-series-1235020378/ |url-status=live}}&lt;/ref&gt; they were followed in July by [[Dana Delany]]&lt;ref&gt;{{cite web |url=https://variety.com/2022/tv/news/taylor-sheridan-tulsa-king-paramount-plus-cast-dana-delany-1235298854/ |title=Taylor Sheridan's 'Tulsa King' Paramount+ Series Casts Dana Delany (Exclusive) |last=Otterson |first=Joe |date=June 20, 2022 |accessdate=February 29, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=March 15, 2023 |archive-url=https://web.archive.org/web/20230315105757/https://variety.com/2022/tv/news/taylor-sheridan-tulsa-king-paramount-plus-cast-dana-delany-1235298854/ |url-status=live}}&lt;/ref&gt; and [[Annabella Sciorra]] in August.&lt;ref&gt;{{cite web |url=https://deadline.com/2022/08/tulsa-king-annabella-sciorra-sylvester-stallone-sister-taylor-sheridan-series-1235092703/ |title='Tulsa King': Annabella Sciorra To Portray Sylvester Stallone's Sister In Taylor Sheridan Series |last=Cordero |first=Rosy |date=August 15, 2022 |accessdate=February 29, 2024 |work=[[Deadline Hollywood]] |archive-date=January 14, 2023 |archive-url=https://web.archive.org/web/20230114070513/https://deadline.com/2022/08/tulsa-king-annabella-sciorra-sylvester-stallone-sister-taylor-sheridan-series-1235092703/ |url-status=live}}&lt;/ref&gt; [[Miles Mussenden]] also stars in the series.&lt;ref&gt;{{cite web |url=https://www.thewrap.com/tulsa-king-cast-miles-mussenden-sylvester-stallone-taylor-sheridan/ |title='Tulsa King': Miles Mussenden Joins Sylvester Stallone in Taylor Sheridan's Mob Drama (Exclusive) |last=Oganesyan |first=Natalie |date=August 11, 2022 |accessdate=February 29, 2024 |work=[[The Wrap]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072014/https://www.thewrap.com/tulsa-king-cast-miles-mussenden-sylvester-stallone-taylor-sheridan/ |url-status=live}}&lt;/ref&gt; Stallone's daughter [[Scarlet Stallone |Scarlet]] appears in the series as Spencer, a barista and stable hand who is recruited to work with Dwight.&lt;ref name=&quot;Scarlet&quot;&gt;{{cite web |url=https://www.usatoday.com/story/entertainment/tv/2022/12/19/tulsa-king-sylvester-stallones-daughter-scarlet-rose-horse-role/10922898002/ |title=Sylvester Stallone's 'most adamant' actress daughter Scarlet Rose rides 'Tulsa King' horse duty |last=Alexander |first=Bryan |date=December 19, 2022 |accessdate=February 29, 2024 |work=[[USA Today]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072013/https://www.usatoday.com/story/entertainment/tv/2022/12/19/tulsa-king-sylvester-stallones-daughter-scarlet-rose-horse-role/10922898002/ |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://variety.com/2023/tv/focus/tv-shows-real-family-members-the-great-tulsa-king-1235645583/ |title='The Great,' 'Tulsa King' and 'Unstable' Casts on the Pros and Cons of Working With Family on Set |last=Saval |first=Malina |date=June 17, 2023 |accessdate=March 3, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=October 27, 2023 |archive-url=https://web.archive.org/web/20231027215945/https://variety.com/2023/tv/focus/tv-shows-real-family-members-the-great-tulsa-king-1235645583/ |url-status=live}}&lt;/ref&gt; Scarlet was originally considered for the role of Stallone's on-screen daughter but was considered too young for the role,&lt;ref name=&quot;Scarlet&quot; /&gt; which instead went to Tatiana Zappardino.&lt;ref&gt;{{cite web |url=https://www.thewrap.com/tulsa-king-tatiana-zappardino-sylvester-stallone/ |title='Tulsa King': Tatiana Zappardino Joins Cast as Sylvester Stallone's Daughter (Exclusive) |last=Oganesyan |first=Natalie |date=August 15, 2022 |accessdate=February 29, 2024 |work=[[The Wrap]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072014/https://www.thewrap.com/tulsa-king-tatiana-zappardino-sylvester-stallone/ |url-status=live}}&lt;/ref&gt; Additional casting for minor characters and background actors occurred in May 2022.&lt;ref&gt;{{cite web |url=https://www.oklahoman.com/story/entertainment/2022/05/18/tulsa-king-sylvester-stallone-movie-filming-okc-releases-november-13/9829592002/ |title=Sylvester Stallone's 'Tulsa King,' now filming in OKC, sets premiere, calls for actors |last=McDonnell |first=Brandy |date=May 18, 2022 |accessdate=February 29, 2024 |work=[[The Oklahoman]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072013/https://www.oklahoman.com/story/entertainment/2022/05/18/tulsa-king-sylvester-stallone-movie-filming-okc-releases-november-13/9829592002/ |url-status=live}}&lt;/ref&gt;

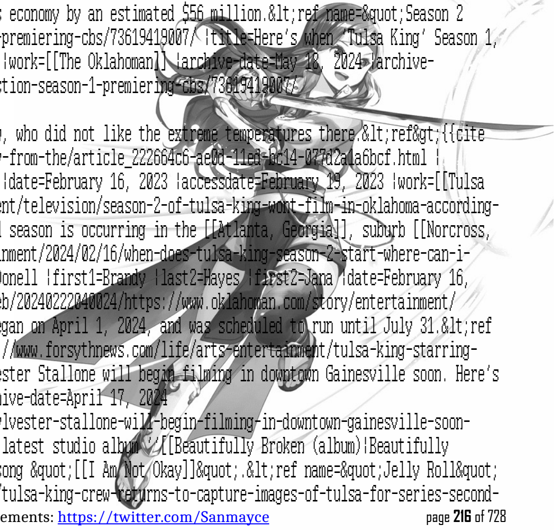
Before production of the second season, Sciorra and Zappardino were promoted to the main cast.&lt;ref&gt;{{cite press release |url=http://www.thefutoncritic.com/news/2024/04/01/production-commences-on-season-two-of-paramountpluss-original-series-tulsa-king-starring-oscar-nominee-sylvester-stallone-476514/20240401cbs01/ |title=Production Commences on Season Two of Paramount+'s Original Series &quot;Tulsa King,&quot; Starring Oscar Nominee Sylvester Stallone |date=April 1, 2024 |accessdate=April 2, 2024 |publisher=[[CBS]] |agency=[[The Futon Critic]] |archive-url=https://ghostarchive.org/archive/20240504050521/http://www.thefutoncritic.com/news/2024/04/01/production-commences-on-season-two-of-paramountpluss-original-series-tulsa-king-starring-oscar-nominee-sylvester-stallone-476514/20240401cbs01/ |archive-date=May 4, 2024 |url-status=live}}&lt;/ref&gt; [[Frank Grillo]] also joined the main cast, starring as Bill Bevilaqua, a mobster from Kansas City.&lt;ref&gt;{{cite web |url=https://deadline.com/2024/04/tulsa-king-frank-grillo-season-2-1235876246/ |title='Tulsa King': Frank Grillo Joins Season 2 Of Paramount+ Series From MTVE Studios |last=Andreeva |first=Nellie |date=April 4, 2024 |accessdate=April 4, 2024 |work=[[Deadline Hollywood]] |archive-date=April 4, 2024 |archive-url=https://web.archive.org/web/20240404211826/https://deadline.com/2024/04/tulsa-king-frank-grillo-season-2-1235876246/ |url-status=live}}&lt;/ref&gt; On May 1, 2024, it was reported [[Neal McDonough]] would appear as Cal Thresher.&lt;ref&gt;{{cite web |url=https://variety.com/2024/tv/news/tulsa-king-neal-mcdonough-season-2-series-regular-1235983634/ |title='Tulsa King' Adds Neal McDonough as Season 2 Series Regular (Exclusive) |last=Longretta |first=Emily |date=May 1, 2024 |accessdate=May 1, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=May 1, 2024 |archive-url=https://web.archive.org/web/20240501143823/https://variety.com/2024/tv/news/tulsa-king-neal-mcdonough-season-2-series-regular-1235983634/ |url-status=live}}&lt;/ref&gt; Two days later, Rich Ting was cast in the recurring role of Jackie Ming.&lt;ref&gt;{{cite web |url=https://deadline.com/2024/05/sylvester-stallone-tulsa-king-rich-ting-cast-season-2-1235904074/ |title=Sylvester Stallone's 'Tulsa King' Casts Rich Ting As Recurring In Season 2 |last=Petski |first=Denise |date=May 3, 2024 |accessdate=May 3, 2024 |work=[[Deadline Hollywood]] |archive-date=May 4, 2024 |archive-url=https://web.archive.org/web/20240504054053/https://deadline.com/2024/05/sylvester-stallone-tulsa-king-rich-ting-cast-season-2-1235904074/ |url-status=live}}&lt;/ref&gt; [[Jelly Roll (singer)|Jelly Roll]], an American singer, songwriter, and rapper, made a [[cameo appearance]] as himself. He said he had been a fan of the series and is friends with [[Sistine Stallone|Sistine]], another daughter of Stallone's, who made the necessary connections.&lt;ref name=&quot;Jelly Roll&quot;&gt;{{cite web|url=https://www.usatoday.com/story/entertainment/tv/2024/09/20/jelly-roll-tulsa-king-appearance/75295227007/|title=Jelly Roll makes 'Tulsa King' TV debut with Sylvester Stallone's mobster: Watch them meet|last=Alexander|first=Bryan|date=September 20, 2024|accessdate=September 22, 2024|work=[[USA Today]]}}&lt;/ref&gt; Casting for additional extras in the second season occurred in June 2024.&lt;ref&gt;{{cite web |url=https://www.okfilmmusic.org/jobs/casting-calls-jobs/extras-needed-next-week-for-tulsa-king-season-2 |title=Extras Needed Next Week for Tulsa King Season 2 |date=June 7, 2024 |accessdate=June 11, 2024 |work=Oklahoma Film Office |archive-url=https://ghostarchive.org/archive/20240612050608/https://www.okfilmmusic.org/jobs/casting-calls-jobs/extras-needed-next-week-for-tulsa-king-season-2 |archive-date=June 12, 2024 |url-status=live }}&lt;/ref&gt;

During filming for the second season, Stallone and an unnamed director were accused of making disparaging remarks about background extras.&lt;ref&gt;{{cite web |url=https://www.avclub.com/sylvester-stallone-tulsa-king-rude-taylor-sheridan-1851397687 |title=Sylvester Stallone's alleged bad behavior pushes extras out of Tulsa King |last=Keates |first=Emma |date=April 9, 2024 |accessdate=April 9, 2024 |work=[[The A.V. Club]] |archive-date=April 9, 2024 |archive-url=https://web.archive.org/web/20240409185317/https://www.avclub.com/sylvester-stallone-tulsa-king-rude-taylor-sheridan-1851397687 |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite magazine |url=https://ew.com/sylvester-stallone-accused-disparaging-tulsa-king-background-actors-8628991 |title=Sylvester Stallone accused of disparaging Tulsa King background actors |last=Wang |first=Jessica |date=April 9, 2024 |accessdate=April 9, 2024 |magazine=[[Entertainment Weekly]] |archive-date=April 9, 2024 |archive-url=https://web.archive.org/web/20240409225301/https://ew.com/sylvester-stallone-accused-disparaging-tulsa-king-background-actors-8628991 |url-status=live}}&lt;/ref&gt; Stallone was allegedly heard calling certain background actors &quot;ugly&quot;, &quot;tub of lard&quot; and &quot;fat guy with a cane&quot;.&lt;ref name=&quot;Rose Locke&quot;&gt;{{cite web |url=https://deadline.com/2024/04/tulsa-king-casting-director-quits-sylvester-stallone-accused-criticizing-background-actors-1235878860/ |title='Tulsa King' Casting Company Quits After Sylvester

Stallone Accused Of Disparaging Background Actors |last=Rice |first=Lynette |date=April 9, 2024 |accessdate=April 9, 2023 |work=[[Deadline Hollywood]] |archive-date=April 9, 2024 |archive-url=https://web.archive.org/web/20240409025312/https://deadline.com/2024/04/tulsa-king-casting-director-quits-sylvester-stallone-accused-criticizing-background-actors-1235878860/ |url-status=live}}&lt;/ref&gt; Stallone also suggested the production should instead cast &quot;pretty young girls to be around me&quot;.&lt;ref&gt;{{cite web |url=https://variety.com/2024/tv/news/sylvester-stallone-tulsa-king-mistreating-extras-1235964689/ |title=Sylvester Stallone Accused of Creating 'Toxic Environment' on Atlanta Set of 'Tulsa King' |last=Aurthur |first=Kate |date=April 9, 2024 |accessdate=April 9, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=April 9, 2024 |archive-url=https://web.archive.org/web/20240409150904/https://variety.com/2024/tv/news/sylvester-stallone-tulsa-king-mistreating-extras-1235964689/ |url-status=live}}&lt;/ref&gt; The Atlanta-based casting agency Rose Locke &amp; CL Casting, which had been responsible for hiring extras, resigned from the production soon after.&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/tv/tv-news/sylvester-stallone-tulsa-king-mocking-ugly-actors-1235869392/ |title=Sylvester Stallone Allegedly Mocked &quot;Ugly&quot; Actors on 'Tulsa King' Set; Director Denies Claims |last=Hibberd |first=James |date=April 9, 2024 |accessdate=April 9, 2024 |work=[[The Hollywood Reporter]] |archive-date=April 9, 2024 |archive-url=https://web.archive.org/web/20240409145456/https://www.hollywoodreporter.com/tv/tv-news/sylvester-stallone-tulsa-king-mocking-ugly-actors-1235869392/ |url-status=live}}&lt;/ref&gt; Zisk responded to the comments through ''[[TMZ]]'' by denying the statements were made and stating the casting agency had hired extras who were older than the requested age range.&lt;ref&gt;{{cite web |url=https://www.tmz.com/2024/04/09/sylvester-stallone-tulsa-king-extras-insults-fat-cane-director/?adid=tw36483276187264 |title=Sylvester Stallone's Tulsa King Actor Accused of Making Disparaging Remarks... Director Denies It Ever Happened |date=April 9, 2024 |accessdate=April 9, 2024 |work=[[TMZ]] |archive-date=April 9, 2024 |archive-url=https://web.archive.org/web/20240409175542/https://www.tmz.com/2024/04/09/sylvester-stallone-tulsa-king-extras-insults-fat-cane-director/?adid=tw36483276187264 |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://people.com/tulsa-king-director-denies-allegations-sylvester-stallone-made-offensive-comments-about-background-actors-8628992 |title=Tulsa King Director Denies Allegations Sylvester Stallone Made Offensive Comments About Background Actors |last=Moore |first=Julia |date=April 9, 2024 |accessdate=April 9, 2024 |work=[[People (magazine) |People]] |archive-date=April 10, 2024 |archive-url=https://web.archive.org/web/20240410035636/https://people.com/tulsa-king-director-denies-allegations-sylvester-stallone-made-offensive-comments-about-background-actors-8628992 |url-status=live}}&lt;/ref&gt; ''[[CNN]]'' reported [[Paramount Global |Paramount]] was investigating the allegations and that no formal complaints had been filed.&lt;ref&gt;{{cite web |url=https://www.cnn.com/2024/04/09/entertainment/sylvester-stallone-tulsa-king/index.html |title=Paramount investigating claims Sylvester Stallone allegedly used disparaging language on 'Tulsa King' set |last=Wagmeister |first=Elizabeth |date=April 9, 2024 |accessdate=April 9, 2024 |work=[[CNN]] |archive-date=April 10, 2024 |archive-url=https://web.archive.org/web/20240410035636/https://www.cnn.com/2024/04/09/entertainment/sylvester-stallone-tulsa-king/index.html |url-status=live}}&lt;/ref&gt; The actors' labor union [[SAG-AFTRA]] also responded to the allegations, saying its scope does not include background actors in Atlanta but that it would provide any requested guidance; it also issued a statement condemning such comments toward any actor.&lt;ref&gt;{{cite web |url=https://variety.com/2024/tv/news/sag-aftra-responds-tulsa-king-allegations-mistreatment-sylvester-stallone-1235967316/ |title=SAG-AFTRA Responds to 'Tulsa King' Allegations: 'There Is No Room on Any Set for Disparaging Comments' |last=Zee |first=Michaela |date=April 10, 2024 |accessdate=May 3, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=May 4, 2024 |archive-url=https://web.archive.org/web/20240504052954/https://variety.com/2024/tv/news/sag-aftra-responds-tulsa-king-allegations-mistreatment-sylvester-stallone-1235967316/ |url-status=live}}&lt;/ref&gt; Thomas Mooneyham, a background actor on the series, stated he believed the comments were about him after he and another extra were replaced with younger people.&lt;ref&gt;{{cite web |url=https://deadline.com/2024/04/tulsa-king-backround-actor-reacts-disparaging-comments-sylvester-stallone-series-1235880947/ |title='Tulsa King' Background Actor Responds To Insults Allegedly Made On Set Of Sylvester Stallone Series: &quot;I Do Feel Like I Was Singled Out&quot; |last=Rice |first=Lynette |date=April 10, 2024 |accessdate=May 3, 2024 |work=[[Deadline Hollywood]] |archive-date=May 4, 2024 |archive-url=https://web.archive.org/web/20240504053739/https://deadline.com/2024/04/tulsa-king-backround-actor-reacts-disparaging-comments-sylvester-stallone-series-1235880947/ |url-status=live}}&lt;/ref&gt; Stallone did not respond to the allegations.&lt;ref&gt;{{cite web |url=https://deadline.com/2024/04/sylvester-stallone-tulsa-king-season-2-announcement-background-actors-allegations-1235886719/ |title=Sylvester Stallone Posts 'Tulsa King' Season 2 Announcement, No Mention Of &quot;Toxic&quot; Set Allegations |last=Tartaglione |first=Nancy |date=April 16, 2024 |accessdate=May 3, 2024 |work=[[Deadline Hollywood]] |archive-date=May 4, 2024 |archive-url=https://web.archive.org/web/20240504054052/https://deadline.com/2024/04/sylvester-stallone-tulsa-king-season-2-announcement-background-actors-allegations-1235886719/ |url-status=live}}&lt;/ref&gt;

3384 The series' costume designer Suzanne McCabe based many outfits on the [[Gambino crime family]] and [[Franzese Crew]], as well as photographs from newspaper clippings in the 1980s; she cited [[John Gotti]] as an inspiration.&lt;ref name=&quot;Costumes&quot;&gt;{{cite web |url=https://variety.com/2023/artisans/awards/tulsa-king-costume-designer-sylvester-stallone-gangster-1235634503/ |title='Tulsa King' Costume Designer Pulled Inspiration From John Gotti and 'Guys and Dolls' for Sylvester Stallone's Look |last=Cernov |first=Matthew |date=June 6, 2023 |accessdate=March 1, 2023 |work=[[Variety (magazine) |Variety]] |archive-date=June 6, 2023 |archive-url=https://web.archive.org/web/20230606181530/https://variety.com/2023/artisans/awards/tulsa-king-costume-designer-sylvester-stallone-gangster-1235634503/ |url-status=live}}&lt;/ref&gt; McCabe also stated she tried to mostly use monochromatic colors for the show's costumes, using dark colors for the [[New York City]]-filmed scenes and softer colors for scenes in Oklahoma to represent that state's [[red soil]].&lt;ref name=&quot;Costumes&quot; /&gt; Sylvester Stallone was allowed to pick many elements of his own costumes, including [[bolo tie]]s, jewelry and shoes.&lt;ref name=&quot;Costumes&quot; /&gt; Production designer Todd Jeffery used a mix of [[soundstage]]s and [[on location filming |on-location filming]] for ''Tulsa King''.&lt;ref name=&quot;Locations&quot;&gt;{{cite web |url=https://variety.com/2023/artisans/awards/tulsa-king-production-design-1235558103/ |title=How 'Tulsa King' Captured the Grungy Side of Oklahoma With Squalid Strip Clubs, Dingy Dive Bars and Ratty Motels |last=Chernov |first=Matthew |date=March 20, 2023 |accessdate=March 2, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=February 1, 2024 |archive-url=https://web.archive.org/web/20240201054135/https://variety.com/2023/artisans/awards/tulsa-king-production-design-1235558103/ |url-status=live}}&lt;/ref&gt; The sets for &quot;The Higher Plane&quot; [[dispensary]] featured in the series was created in a former [[Texaco]] gas station.&lt;ref name=&quot;Locations&quot; /&gt; Mirrors were used in a scene that was filmed in a [[strip club]] to make the space appear larger.&lt;ref name=&quot;Locations&quot; /&gt; Location manager Patrick Mignanom was tasked with finding a dilapidated structure whose owner would be content with having it blown up.&lt;ref name=&quot;Locations&quot; /&gt; Saunder Jurriaans and Danny Bensi composed ''Tulsa King''{{'}}s theme song.&lt;ref&gt;{{cite web |url=https://listen.tidal.com/album/264651446/credits |title=Credits/Tulsa King (Original Theme)/Danny Bensi |accessdate=May 4, 2024 |work=[[Tidal (service) |Tidal]] |archive-url=https://web.archive.org/web/20240504062436/https://listen.tidal.com/album/264651446/credits |archive-date=May 4, 2024 |url-status=live}}&lt;/ref&gt;

3385 [[Principal photography]] occurred over six months in [[Oklahoma City]], [[Tulsa, Oklahoma |Tulsa]] and [[Bethany, Oklahoma |Bethany]], concluding on August 31, 2022.&lt;ref name=&quot;Filming dates&quot;&gt;{{Cite web |date=October 25, 2022 |title=Paramount wraps season one of Tulsa King |url=https://okfilmmusic.org/news/paramount-wraps-season-one-of-tulsa-king-in-oklahoma-ahead-of-november-premiere |access-date=November 13, 2022 |website=Oklahoma Film Office |archive-date=March 24, 2023 |archive-url=https://web.archive.org/web/20230324093538/https://okfilmmusic.org/news/paramount-wraps-season-one-of-tulsa-king-in-oklahoma-ahead-of-november-premiere |url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://deadline.com/2022/08/tulsa-king-sylvester-stallone-production-wrapped-oklahoma-november-debut-1235104419/ |title='Tulsa King': Sylvester Stallone Says &quot;I Never Worked So Hard In My Life&quot; |last=Rice |first=Lynette |date=August 31, 2022 |accessdate=February 29, 2024 |work=[[Deadline Hollywood]] |archive-date=October 27, 2022 |archive-url=https://web.archive.org/web/20221027125546/https://deadline.com/2022/08/tulsa-king-sylvester-stallone-production-wrapped-oklahoma-november-debut-1235104419/ |url-status=live}}&lt;/ref&gt; Some filming occurred at [[Tulsa International Airport]] on March 29, 2022.&lt;ref&gt;{{cite web |url=https://tulsaworld.com/entertainment/television/sylvester-stallone-shoots-tulsa-king-scenes-at-tulsa-international-airport/article_c60b0800-af60-11ec-96c8-a7e1f8a3ac4c.html |title=Sylvester Stallone shoots 'Tulsa King' scenes at Tulsa International Airport |last=Tramel |first=Jimmie |date=March 30, 2022 |accessdate=February 29, 2024 |work=[[Tulsa World]] |archive-date=May 1, 2022 |archive-url=https://web.archive.org/web/20220501024039/https://tulsaworld.com/entertainment/television/sylvester-stallone-shoots-tulsa-king-scenes-at-tulsa-international-airport/article_c60b0800-af60-11ec-96c8-a7e1f8a3ac4c.html |url-status=live}}&lt;/ref&gt; Additional locations used in Tulsa include [[List_of_places_referred_to_as_the_Center_of_the_Universe#United_States |Center of the Universe]] and the [[Mayo Hotel]].&lt;ref&gt;{{cite web |url=https://www.visittulsa.com/blog/post/tulsa-king-filming-locations/ |title=Explore These 'Tulsa King' Filming Locations |date=November 11, 2022 |accessdate=February 29, 2024 |work=Visit Tulsa |archive-date=January 12, 2024 |archive-url=https://web.archive.org/web/20240112162710/https://www.visittulsa.com/blog/post/tulsa-king-filming-locations/ |url-status=live}}&lt;/ref&gt; Other scenes were filmed on-location in [[Brooklyn, New York]].&lt;ref&gt;{{cite web |url=https://people.com/movies/sylvester-stallone-spends-time-daughter-sophia-while-filming-in-nyc/ |title=Sylvester Stallone Spends Time with Daughter Sophia While Filming 'Tulsa King' in N.Y.C. |last=Lindert |first=Hattie |date=May 18, 2022 |accessdate=February 29, 2024 |work=[[People (magazine) |People]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072015/https://people.com/movies/sylvester-stallone-spends-time-daughter-sophia-while-filming-in-nyc/ |url-status=live}}&lt;/ref&gt; The series interior scenes and production offices were housed at [[Prairie Surf Studios]].&lt;ref&gt;{{cite web |url=https://www.oklahoman.com/story/entertainment/2024/02/28/tulsa-king-season-1-debut-cbs-how-to-watch/72779920007/ |title='Tulsa King' Season 1, filmed largely in Oklahoma, to debut on CBS this summer: What to know |last=McDonnell |first=Brandy |date=February 28, 2024 |accessdate=February 29, 2024 |work=[[The Oklahoman]] |archive-date=February 29, 2024 |archive-url=https://web.archive.org/web/20240229072014/https://www.oklahoman.com/story/entertainment/2024/02/28/tulsa-king-season-1-debut-cbs-how-to-watch/72779920007/ |url-status=live}}&lt;/ref&gt; [[Additional photography]] wrapped by October.&lt;ref name=&quot;Filming dates&quot; /&gt; Filming in Oklahoma boosted the state's economy by an estimated $56 million.&lt;ref name=&quot;Season 2 filming&quot;&gt;{{cite web |url=https://www.oklahoman.com/story/entertainment/2024/05/09/tulsa-king-season-2-production-season-1-premiering-cbs/73619419007/ |title=Here's when 'Tulsa King' Season 1, filmed in Oklahoma, will premiere on network television |last=McDonnell |first=Brandy |date=May 9, 2024 |accessdate=June 11, 2024 |work=[[The Oklahoman]] |archive-date=May 18, 2024 |archive-url=https://web.archive.org/web/20240518210239/https://www.oklahoman.com/story/entertainment/2024/05/09/tulsa-king-season-2-production-season-1-premiering-cbs/73619419007/ |url-status=live}}&lt;/ref&gt;



3386 It was later reported the show's second season would not be filmed in Oklahoma due to costs, and complaints from the cast and crew, who did not like the extreme temperatures there.&lt;ref&gt;{{cite web |url=https://tulsaworld.com/entertainment/television/season-2-of-tulsa-king-wont-film-in-oklahoma-according-to-exclusive-story-from-the/article_222664c6-ae0d-11ed-bc14-077d2a1a6bcf.html |title=Season 2 of 'Tulsa King' won't film in Oklahoma, according to exclusive story from The Oklahoman |last=Tramel |first=Jimmie |date=February 16, 2023 |accessdate=February 19, 2023 |work=[[Tulsa World]] |archive-date=February 19, 2023 |archive-url=https://web.archive.org/web/20230219231815/https://tulsaworld.com/entertainment/television/season-2-of-tulsa-king-wont-film-in-oklahoma-according-to-exclusive-story-from-the/article_222664c6-ae0d-11ed-bc14-077d2a1a6bcf.html |url-status=live}}&lt;/ref&gt; Filming on the second season is occurring in the [[Atlanta, Georgia]], suburb [[Norcross, Georgia |Norcross]], using Eagle Rock Studios for sound stages.&lt;ref&gt;{{cite web |url=https://www.oklahoman.com/story/entertainment/2024/02/16/when-does-tulsa-king-season-2-start-where-can-i-watch-tulsa-king/72626704007/ |title=What changes are coming for 'Tulsa King' Season 2? What to know about release date |last1=McDonell |first1=Brandy |last2=Hayes |first2=Jana |date=February 16, 2024 |accessdate=February 21, 2024 |work=[[The Oklahoman]] |archive-date=February 22, 2024 |archive-url=https://web.archive.org/web/20240222040024/https://www.oklahoman.com/story/entertainment/2024/02/16/when-does-tulsa-king-season-2-start-where-can-i-watch-tulsa-king/72626704007/ |url-status=live}}&lt;/ref&gt; Filming began on April 1, 2024, and was scheduled to run until July 31.&lt;ref name=&quot;Season 2 filming&quot; /&gt; Some filming also took place in [[Gainesville, Georgia]].&lt;ref&gt;{{cite web |url=https://www.forsythnews.com/life/arts-entertainment/tulsa-king-starring-sylvester-stallone-will-begin-filming-in-downtown-gainesville-soon-heres-what-you-need-to-know/ |title='Tulsa King' starring Sylvester Stallone will begin filming in downtown Gainesville soon. Here's what you need to know |last=Anderson |first=Ben |date=April 10, 2024 |accessdate=June 11, 2024 |work=[[Forsyth County News]] |archive-date=April 17, 2024 |archive-url=https://web.archive.org/web/20240417155929/https://www.forsythnews.com/life/arts-entertainment/tulsa-king-starring-sylvester-stallone-will-begin-filming-in-downtown-gainesville-soon-heres-what-you-need-to-know/ |url-status=live}}&lt;/ref&gt; Jelly Roll's scenes were filmed in April while he was touring for his latest studio album ''[[Beautifully Broken (album)|Beautifully Broken]]'' and was in the area. Stallone improvised lines for this scene which featured Jelly Roll performing a rendition of his song &quot;[[I Am Not Okay]]&quot;.&lt;ref name=&quot;Jelly Roll&quot; /&gt; A [[second unit]] filmed [[b-roll]] scenes in Tulsa in mid-June 2024.&lt;ref&gt;{{cite web |url=https://tulsaworld.com/news/tulsa-king-crew-returns-to-capture-images-of-tulsa-for-series-second-season/article_08714942-2823-11ef-9463-1baf58cf17ba.html |title='Tulsa King' crew returns to capture images of Tulsa for series' second season |last1=Tramel |first1=Jimmie |last2=Horner |first2=Taylor |date=June 11, 2024 |accessdate=June 11, 2024 |work=[[Tulsa World]] |url-access=subscription |archive-date=June 12, 2024 |archive-url=https://web.archive.org/web/20240612032003/https://tulsaworld.com/news/tulsa-king-crew-returns-to-capture-images-of-tulsa-for-series-second-season/article_08714942-2823-11ef-9463-1baf58cf17ba.html |url-status=live}}&lt;/ref&gt; Filming for the second season wrapped on August 2.&lt;ref&gt;{{cite instagram|postid=C-K5SZ4JVDu|title=That's a wrap on Tulsa King season 2. Thank you to the incredible crew who made this possible through hell and high water. It was a challenge that we overcame together, and I'm so grateful to have these soldiers by my side. Check out season 2 of Tulsa King out September 15 on @paramountplus.|user=officialslystallone|last=Stallone|first=Sylvester|author-link=Sylvester Stallone|date=August 2, 2024|accessdate=August 4, 2024}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://screenrant.com/tulsa-king-season-2-filming-update-sylvester-stallone-wrap-video/|title=Tulsa King Season 2 Wraps Filming As Sylvester Stallone Shares BTS Video|last=Northrup|first=Ryan|date=August 2, 2024|accessdate=August 4, 2024|work=[[ScreenRant]]}}&lt;/ref&gt;

The series premiere episode of ''Tulsa King'' received an early promotional screening with the fifth-season premiere of ''[[Yellowstone (American TV series) |Yellowstone]]'' in [[AMC Theatres]] on October 29-30, 2022.&lt;ref&gt;{{cite web |url=https://variety.com/2022/tv/news/yellowstone-season-5-premiere-tulsa-king-amc-theaters-1235409589/ |title='Yellowstone' Season 5 Premiere, 'Tulsa King' Debut to Get Special Preview in AMC Theaters |last=Otterson |first=Joe |date=October 20, 2022 |accessdate=February 29, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=November 4, 2022 |archive-url=https://web.archive.org/web/20221104013232/https://variety.com/2022/tv/news/yellowstone-season-5-premiere-tulsa-king-amc-theaters-1235409589/ |url-status=live}}&lt;/ref&gt;&lt;ref&gt;{{cite web |url=https://www.amctheatres.com/amc-scene/yellowstone-and-tulsa-king-premiere-early-on-the-big-screen |title=Yellowstone And Tulsa King Premiere Early On The Big Screen |date=October 26, 2022 |accessdate=March 1, 2024 |work=[[AMC Theatres]] |archive-date=January 17, 2023 |archive-url=https://web.archive.org/web/20230117204713/https://www.amctheatres.com/amc-scene/yellowstone-and-tulsa-king-premiere-early-on-the-big-screen |url-status=live}}&lt;/ref&gt; ''Tulsa King'' began its weekly release schedule on [[Paramount+]] beginning on November 13.&lt;ref&gt;{{cite web |url=https://tvline.com/lists/tv-schedule-what-to-watch-sylvester-stallone-tulsa-king-paramount-plus/sunday-november-13-2022/ |title=The TVLine-Up: What's New, Returning and Leaving the Week of Nov. 13 |last=Schwartz |first=Ryan |date=November 12, 2022 |accessdate=February 29, 2024 |work=[[TVLine]] |archive-date=November 14, 2022 |archive-url=https://web.archive.org/web/20221114142951/https://tvline.com/lists/tv-schedule-what-to-watch-sylvester-stallone-tulsa-king-paramount-plus/sunday-november-13-2022 |url-status=live}}&lt;/ref&gt; The series' first two episodes were [[linear broadcasting |broadcast]] on [[Paramount Network]] on November 20 and 27, serving as a lead-out for episodes of ''Yellowstone''.&lt;ref&gt;{{cite web |url=https://deadline.com/2022/05/sylvester-stallone-tulsa-king-paramount-premiere-date-paramount-plus-yellowstone-1235027354/ |title=Sylvester Stallone's 'Tulsa King' Gets Paramount+ Premiere Date, Simultaneous Cable Launch |last=Pedersen |first=Erik |date=May 18, 2022 |accessdate=February 29, 2024 |work=[[Deadline Hollywood]] |archive-date=February 2, 2023 |archive-url=https://web.archive.org/web/20230202111730/https://deadline.com/2022/05/sylvester-stallone-tulsa-king-paramount-premiere-date-paramount-plus-yellowstone-1235027354/ |url-status=live}}&lt;/ref&gt; The first season concluded on January 8, 2023.&lt;ref&gt;{{cite web |url=https://tvline.com/what-to-watch/tulsa-king-season-1-finale-episode-9-paramount-plus/ |title=Tulsa King Finale, Mayfair Witches Premiere, Scott Caan's Alert Preview, East New York and More |date=January 8, 2023 |accessdate=February 29, 2024 |work=[[TVLine]] |archive-date=February 7, 2023 |archive-url=https://web.archive.org/web/20230207011501/https://tvline.com/what-to-watch/tulsa-king-season-1-finale-episode-9-paramount-plus/ |url-status=live}}&lt;/ref&gt; In Japan, the series was launched in a Paramount+ hub on the streaming service [[Wowow]].&lt;ref&gt;{{cite web |url=https://variety.com/2023/tv/news/paramount-to-launch-in-japan-1235728416/ |title=Paramount+ to Launch in Japan via Partnerships With J:COM and Wowow |last=Frater |first=Patrick |date=September 19, 2023 |accessdate=February 29, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=November 16, 2023 |archive-url=https://web.archive.org/web/20231116043447/https://variety.com/2023/tv/news/paramount-to-launch-in-japan-1235728416/ |url-status=live}}&lt;/ref&gt; The first season received another linear broadcast, this time on [[CBS]], with episodes being broadcast weekly from July 14-September 8, 2024.&lt;ref&gt;{{cite web |url=https://deadline.com/2024/05/cbs-summer-schedule-when-to-expect-big-brother-tonys-tulsa-king-network-debut-1235907593/ |title=CBS Summer Schedule: When To Expect 'Big Brother,' Tonys &amp; 'Tulsa King' Network Debut |last=Rice\first=Lynette |date=May 7, 2024 |accessdate=June 11, 2024 |work=[[Deadline Hollywood]] |archive-date=May 24, 2024 |archive-url=https://web.archive.org/web/20240524212830/https://deadline.com/2024/05/cbs-summer-schedule-when-to-expect-big-brother-tonys-tulsa-king-network-debut-1235907593/ |url-status=live}}&lt;/ref&gt;&lt;ref name=&quot;TFC&quot; /&gt; The second season premiered in the United States and Canada on September 15, and internationally on September 16.&lt;ref&gt;{{cite web |url=https://deadline.com/2024/06/tulsa-king-season-2-premiere-date-sylvester-stallone-trailer-1235983665/ |title='Tulsa King' Sets Season 2 Premiere Date; Sylvester Stallone Is Behind Bars In First Trailer |last=Petski |first=Denise |date=June 26, 2024 |accessdate=June 26, 2024 |work=[[Deadline Hollywood]]}}&lt;/ref&gt;

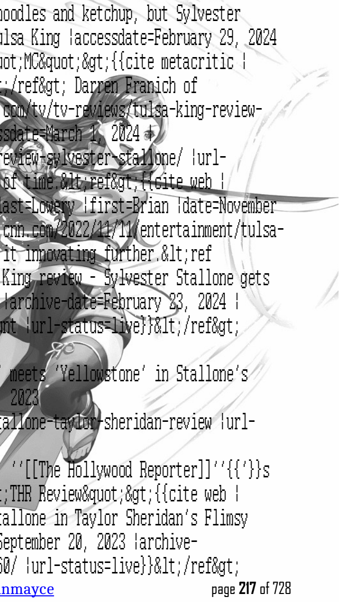
| region1 = June 6, 2023&lt;ref name=&quot;Home media release&quot;&gt;{{cite web |url=https://okcfox.com/news/entertainment/tulsa-king-sylvester-stallone-paramount-dvd-bluray-blu-ray-june-6-2023-dwight-manfredi-new-york-oklahoma-city-home-media-entertainment-steelbook-streaming-series-release-date-gangsters-domenick-lombardozzi-vincent-piazza-max-casella |title=Paramount to release Sylvester Stallone series 'Tulsa King' on DVD and Blu-ray |date=March 31, 2023 |accessdate=May 3, 2024 |work=[[KOKH]] |archive-date=June 2, 2023 |archive-url=https://web.archive.org/web/20230602202355/https://okcfox.com/news/entertainment/tulsa-king-sylvester-stallone-paramount-dvd-bluray-blu-ray-june-6-2023-dwight-manfredi-new-york-oklahoma-city-home-media-entertainment-steelbook-streaming-series-release-date-gangsters-domenick-lombardozzi-vincent-piazza-max-casella |url-status=live}}&lt;/ref&gt;

On [[Paramount+]], ''Tulsa King''{{'}}s first season was [[Taylor Sheridan]]'s fourth-most-watched series&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/tv/tv-news/taylor-sheridan-tulsa-king-starring-sylvester-stallone-to-air-on-cbs-1235838276/ |title=Taylor Sheridan's 'Tulsa King,' Starring Sylvester Stallone, to Air on CBS |last=Goldberg |first=Lesley |date=February 28, 2024 |accessdate=February 29, 2024 |work=[[The Hollywood Reporter]] |archive-date=February 28, 2024 |archive-url=https://web.archive.org/web/20240228215811/https://www.hollywoodreporter.com/tv/tv-news/taylor-sheridan-tulsa-king-starring-sylvester-stallone-to-air-on-cbs-1235838276/ |url-status=live}}&lt;/ref&gt; with 3.36 billion minutes viewed.&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/tv/tv-news/most-watched-taylor-sheridan-tv-shows-yellowstone-more-1235836003/ |title=Taylor Sheridan's Most (and Least) Watched TV Shows, Ranked |last1=Hibberd |first1=James |last2=Porter |first2=Rick |date=February 27, 2024 |accessdate=February 29, 2024 |work=[[The Hollywood Reporter]] |archive-date=February 28, 2024 |archive-url=https://web.archive.org/web/20240228210928/https://www.hollywoodreporter.com/tv/tv-news/most-watched-taylor-sheridan-tv-shows-yellowstone-more-1235836003/ |url-status=live}}&lt;/ref&gt; In its first broadcast on [[Paramount Network]], the series brought in 3.7 million viewers, exceeding the ''[[Game of Thrones]]'' prequel ''[[House of the Dragon]]'' as &quot;cable's highest-rated series debut&quot; of 2022.&lt;ref&gt;{{cite web |url=https://www.hollywoodreporter.com/tv/tv-news/tulsa-king-season-2-ratings-1235272413/ |title='Tulsa King' Renewed for Season 2, Premiere Ratings Top 'House of the Dragon' |last=Hibberd |first=James |date=November 30, 2022 |accessdate=February 29, 2024 |work=[[The Hollywood Reporter]] |archive-date=February 4, 2023 |archive-url=https://web.archive.org/web/20230204174638/https://www.hollywoodreporter.com/tv/tv-news/tulsa-king-season-2-ratings-1235272413/ |url-status=live}}&lt;/ref&gt; It is also credited with the &quot;biggest new sign-up day in [Paramount+] history&quot;.&lt;ref&gt;{{cite web |url=https://variety.com/2022/tv/news/tulsa-king-renewed-season-2-paramount-plus-1235445674/ |title='Tulsa King' Renewed for Season 2 at Paramount+ |last=Otterson |first=Joe |date=November 30, 2022 |accessdate=February 29, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=February 5, 2023 |archive-url=https://web.archive.org/web/20230205160224/https://variety.com/2022/tv/news/tulsa-king-renewed-season-2-paramount-plus-1235445674/ |url-status=live}}&lt;/ref&gt; The first season broadcast on CBS averaged {{Avg|round=2|3.398|3.459|2.682|2.750|2.967|3.426|3.583|2.721|3.508}} million viewers.&lt;ref name=&quot;Linear broadcast ratings&quot;&gt;Attributed to multiple references:

* For episode 9, see: {{cite web|url=https://programminginsider.com/sunday-ratings-nfl-kickoff-weekend-scores-big-numbers-for-fox-cbs-and-nbc/|title=Sunday Ratings: NFL Kickoff Weekend Scores Big Numbers for Fox, CBS and NBC|last=Pucci|first=Douglas|date=September 10, 2024|accessdate=September 29, 2024|work=Programming Insider}}&lt;/ref&gt; The second season premiere on Paramount+ was seen by two million people within 24-hours, with the number rising to 5.4 million households within its first seven days.&lt;ref name=&quot;S2 viewers&quot;&gt;{{cite web|url=https://deadline.com/2024/09/tulsa-king-season-2-premiere-ratings-sylvester-stallone-1236100512/|title='Tulsa King': Sylvester Stallone Returns To Record Audience With Season 2 Premiere|last=Campione|first=Katie|date=September 26, 2024|accessdate=September 29, 2024|work=[[Deadline Hollywood]]}}&lt;/ref&gt; This also set a new record on the streaming service for largest number of viewers on a series' premiere day.&lt;ref&gt;{{cite press release|url=http://www.thefutoncritic.com/ratings/2024/09/26/tulsa-king-packs-a-punch-with-season-two-premiere-streamed-by-54m-global-households-86312/20240926cbs03/|title=&quot;Tulsa King&quot; Packs a Punch with Season Two Premiere Streamed by 5.4M Global Households|publisher=Paramount Press Express|via=[[The Futon Critic]]|date=September 26, 2024|accessdate=September 29, 2024}}&lt;/ref&gt; Social media engagements for season two also rose 943% when compared to that from the first season.&lt;ref name=&quot;S2 viewers&quot; /&gt; ''Tulsa King'' was ranked tenth on ''[[The Wrap]]''{{'}}s &quot;list of titles consumers are most excited about&quot; for the week of September 22, 2024.&lt;ref&gt;{{cite web|url=https://www.thewrap.com/tulsa-king-season-2-premiere-excitement-chart/|title='Tulsa King' Muscles Onto List of Titles Consumers Are Most Excited About After Season 2 Premiere|date=September 25, 2024|accessdate=September 29, 2024|work=[[The Wrap]]|url-access=subscription|archive-url=https://archive.ph/1WFp5|archive-date=September 29, 2024|url-status=live}}&lt;/ref&gt;

{{rotten tomatoes prose|79 |6.6 |47 |consensus=''Tulsa King''{{'}}s stale comedy sometimes feels like ordering spaghetti with marinara and instead getting egg noodles and ketchup, but Sylvester Stallone still commands the screen with his swaggering charm.}}&lt;ref name=&quot;RT&quot;&gt;{{cite rotten tomatoes |id=tulsa_king |type=tv |season=1 |title=Tulsa King |accessdate=February 29, 2024 |archive-url=https://ghostarchive.org/archive/tOgcc |archive-date=March 2, 2024 |url-status=live}}&lt;/ref&gt; {{metacritic film prose |65 |28}}&lt;ref name=&quot;MC&quot;&gt;{{cite metacritic |id=tulsa-king |type=tv |season=1 |accessdate=March 1, 2024 |archive-url=https://ghostarchive.org/archive/FaQWG |archive-date=March 2, 2024 |url-status=live}}&lt;/ref&gt; Darren Franich of ''[[Entertainment Weekly]]'' described the series as &quot;Grumpy Old ''Grand Theft Auto''&quot;.&lt;ref name=&quot;EW&quot;&gt;{{cite magazine |url=https://ew.com/tv/tv-reviews/tulsa-king-review-sylvester-stallone/ |title=Tulsa King review: Sylvester Stallone stars in Grumpy Old Grand Theft Auto |last=Franich |first=Darren |date=November 11, 2022 |accessdate=March 1, 2024 |magazine=[[Entertainment Weekly]] |archive-date=April 19, 2023 |archive-url=https://web.archive.org/web/20230419215440/https://ew.com/tv/tv-reviews/tulsa-king-review-sylvester-stallone/ |url-status=live}}&lt;/ref&gt; ''[[CNN Entertainment]]''{{'}}s Brian Lowery described the series as an &quot;odd mix of attributes&quot; but praised the series' use of time.&lt;ref&gt;{{cite web |url=https://www.cnn.com/2022/11/11/entertainment/tulsa-king-review/index.html |title=Sylvester Stallone tries to rewind the clock in Paramount's 'Tulsa King' |last=Lowery |first=Brian |date=November 11, 2022 |accessdate=March 1, 2024 |work=[[CNN Entertainment]] |archive-date=April 15, 2023 |archive-url=https://web.archive.org/web/20230415042505/https://www.cnn.com/2022/11/11/entertainment/tulsa-king-review/index.html |url-status=live}}&lt;/ref&gt; Reviewing for ''[[The Guardian]]'', Lucy Mangan praised the comedy aspect of the series but failed to see it innovating further.&lt;ref name=&quot;TG&quot;&gt;{{cite web |url=https://www.theguardian.com/tv-and-radio/2022/nov/15/tulsa-king-review-sylvester-stallone-mobster-paramount |title=Tulsa King review – Sylvester Stallone gets his first ever TV role! As a slow-moving 75-year-old gangster |last=Mangan |first=Lucy |date=November 15, 2022 |accessdate=March 1, 2024 |work=[[The Guardian]] |archive-date=February 23, 2024 |archive-url=https://web.archive.org/web/20240223212615/https://www.theguardian.com/tv-and-radio/2022/nov/15/tulsa-king-review-sylvester-stallone-mobster-paramount |url-status=live}}&lt;/ref&gt; ''[[Los Angeles Times]]'' writer Robert Lloyd called the series &quot;likeable&quot;, commending its use of comedy and character focus.&lt;ref&gt;{{cite web |url=https://www.latimes.com/entertainment-arts/tv/story/2022-11-13/tulsa-king-paramount-network-sylvester-stallone-taylor-sheridan-review |title='The Sopranos' meets 'Yellowstone' in Stallone's likable crime comedy 'Tulsa King' |last=Lloyd |first=Robert |date=November 13, 2022 |accessdate=March 1, 2023 |work=[[Los Angeles Times]] |archive-date=April 4, 2023 |archive-url=https://web.archive.org/web/20230404154113/https://www.latimes.com/entertainment-arts/tv/story/2022-11-13/tulsa-king-paramount-network-sylvester-stallone-taylor-sheridan-review |url-status=live}}&lt;/ref&gt;

''Tulsa King'' was often compared poorly to Winter's and Sheridan's other series; Sheridan was overseeing eight other series at the time ''Tulsa King'' debuted. ''[[The Hollywood Reporter]]''{{'}}s Daniel Fienberg wrote Sheridan's and Winter's main strength is not in comedy writing, despite the series being primarily marketed as a comedy.&lt;ref name=&quot;THR Review&quot;&gt;{{cite web |url=https://www.hollywoodreporter.com/tv/tv-reviews/tulsa-king-sylvester-stallone-taylor-sheridan-paramount-1235259060/ |title='Tulsa King' Review: Sylvester Stallone in Taylor Sheridan's Flimsy Fish-Out-of-Water Mobster Series |last=Fienberg |first=Daniel |date=November 11, 2022 |accessdate=March 1, 2024 |work=[[The Hollywood Reporter]] |archive-date=September 20, 2023 |archive-url=https://web.archive.org/web/20230920170646/https://www.hollywoodreporter.com/tv/tv-reviews/tulsa-king-sylvester-stallone-taylor-sheridan-paramount-1235259060/ |url-status=live}}&lt;/ref&gt;



Fienberg goes on to state: &quot;the first two episodes definitely give the impression of being something that Sheridan, Paramount+'s golden goose at this point, gestated between work on 15 different ''Yellowstone'' sequels and prequels&quot;.&lt;ref name=&quot;THR Review&quot; /&gt; According to Anita Singh of ''[[The Daily Telegraph]]'': &quot;One of the writers, Terence Winter, has ''The Sopranos'' and ''Boardwalk Empire'' on his CV, but this show is to ''The Sopranos'' what ''Paw Patrol'' is to the works of David Attenborough&quot;.&lt;ref name=&quot;TDT&quot;&gt;{{cite web |url=https://www.telegraph.co.uk/tv/2022/11/11/tulsa-king-review-even-sylvester-stallone-cant-save-outdated/ |title=Tulsa King, review: even Sylvester Stallone can't save this outdated Mafia comedy |last=Singh |first=Anita |date=November 11, 2022 |accessdate=March 1, 2024 |work=[[The Daily Telegraph]] |archive-date=March 1, 2024 |archive-url=https://web.archive.org/web/20240301070930/https://www.telegraph.co.uk/tv/2022/11/11/tulsa-king-review-even-sylvester-stallone-cant-save-outdated/ |url-status=live}}&lt;/ref&gt;

3393 Writing for the [[National Public Radio]] (NPR) talk show ''[[Fresh Air]]'', David Bianculli compared the formula of ''Tulsa King'' to that of ''Yellowstone'' and noted the &quot;sense of time passing and the importance of family&quot;.&lt;ref&gt;{{cite web |url=https://www.npr.org/2022/11/10/1135789679/movie-stars-head-to-tv-in-the-winning-dramas-yellowstone-and-tulsa-king |title=Movie stars head to TV in the winning dramas 'Yellowstone' and 'Tulsa King' |last=Bianculli |first=David |date=November 10, 2022 |accessdate=March 1, 2024 |work=[[Fresh Air]] |publisher=[[National Public Radio]] |archive-date=June 17, 2023 |archive-url=https://web.archive.org/web/20230617083845/https://www.npr.org/2022/11/10/1135789679/movie-stars-head-to-tv-in-the-winning-dramas-yellowstone-and-tulsa-king |url-status=live}}&lt;/ref&gt; Stephan Lee with ''[[The Wrap]]'' said there is a &quot;distinct straight-to-DVD quality to ''Tulsa King''&quot; but suggests the series stands on its own and is only poor when compared to Sheridan's and Winter's other works.&lt;ref&gt;{{cite web |url=https://www.thewrap.com/tulsa-king-review-sylvester-stallone-paramount-plus-show-taylor-sheridan/ |title='Tulsa King' Review: Stallone Embraces the Absurd in Quirky Taylor Sheridan Series |last=Lee |first=Stephan |date=November 13, 2022 |accessdate=March 1, 2024 |work=[[The Wrap]] |archive-date=June 2, 2023 |archive-url=https://web.archive.org/web/20230602173402/https://www.thewrap.com/tulsa-king-review-sylvester-stallone-paramount-plus-show-taylor-sheridan/ |url-status=live}}&lt;/ref&gt; Ben Travers of ''[[IndieWire]]'' described ''Tulsa King'' as &quot;less serious&quot; and a &quot;breath of fresh air&quot; compared to the seriousness and consequences in Sheridan's other series.&lt;ref name=&quot;IW&quot;&gt;{{cite web |url=https://www.indiewire.com/criticism/shows/tulsa-king-review-sylvester-stallone-show-1234780852/ |title=Sylvester Stallone's First TV Show, 'Tulsa King,' Is All in Good Fun |last=Travers |first=Ben |date=November 11, 2022 |accessdate=March 1, 2024 |work=[[IndieWire]] |archive-date=February 3, 2024 |archive-url=https://web.archive.org/web/20240203221402/https://www.indiewire.com/criticism/shows/tulsa-king-review-sylvester-stallone-show-1234780852/ |url-status=live}}&lt;/ref&gt;

3394 Stallone's acting received the highest praise from critics, although some criticized it for its lack of originality. Reviewing the first two episodes for ''[[Variety (magazine) |Variety]]'', Joshua Alston credited most of the series' success to Stallone, stating: &quot;''Tulsa King'' isn't a great show with him, but it would be far less interesting without him&quot;.&lt;ref&gt;{{cite web |url=https://variety.com/2022/tv/reviews/tulsa-king-review-taylor-sheridan-sylvester-stallone-1235429631/ |title='Tulsa King' Is a Rickety Star Vehicle for Sylvester Stallone: TV Review |last=Alston |first=Joshua |date=November 11, 2022 |accessdate=March 1, 2024 |work=[[Variety (magazine) |Variety]] |archive-date=November 3, 2023 |archive-url=https://web.archive.org/web/20231103075155/https://variety.com/2022/tv/reviews/tulsa-king-review-taylor-sheridan-sylvester-stallone-1235429631/ |url-status=live}}&lt;/ref&gt; Richard Roeper, writing for the ''[[Chicago Sun-Times]]'', said the series was written to Stallone's strengths.&lt;ref name=&quot;CST&quot;&gt;{{cite web |last=Roeper |first=Richard |date=November 11, 2022 |title='Tulsa King': Stallone doin' fine in Oklahoma as a transplanted N.Y. mobster |url=https://chicago.suntimes.com/movies-and-tv/2022/11/11/23451411/tulsa-king-review-sylvester-stallone-paramount-series |url-status=live |archive-url=https://web.archive.org/web/20231103073710/https://chicago.suntimes.com/movies-and-tv/2022/11/11/23451411/tulsa-king-review-sylvester-stallone-paramount-series |archive-date=November 3, 2023 |accessdate=March 1, 2024 |work=[[Chicago Sun-Times]]}}&lt;/ref&gt; The ''[[A.V. Club]]''{{'}}s Todd Lazarski also praised Stallone's acting but described the series as an &quot;undercooked fish-out-of-water mob story&quot;.&lt;ref name=&quot;AVC&quot;&gt;{{cite web |url=https://www.avclub.com/tulsa-king-review-tv-paramount-plus-sylvester-stallone-1849768956 |title=Sylvester Stallone tries the whole TV thing with Tulsa King |last=Lazarski |first=Todd |date=November 11, 2022 |accessdate=March 1, 2024 |work=[[The A.V. Club]] |archive-date=March 27, 2023 |archive-url=https://web.archive.org/web/20230327141706/https://www.avclub.com/tulsa-king-review-tv-paramount-plus-sylvester-stallone-1849768956 |url-status=live}}&lt;/ref&gt; ''[[USA Today]]'' writer Kelly Lawler criticized both the overall concept and Stallone saying he is &quot;probably the king of something, but it's certainly not Tulsa, Oklahoma&quot;, and called the show &quot;bad ''Goodfellas'' fan fiction&quot;.&lt;ref name=&quot;USAT&quot;&gt;{{cite web |url=https://www.usatoday.com/story/entertainment/tv/2022/11/11/tulsa-king-review-sylvester-stallone-first-tv-show-mobster-mess/8323288001/ |title=Review: Sylvester Stallone's 'Tulsa King' is bad 'Goodfellas' fan fiction |last=Lawler |first=Kelly |date=November 11, 2022 |accessdate=March 1, 2024 |work=[[USA Today]] |archive-date=June 1, 2023 |archive-url=https://web.archive.org/web/20230601052844/https://www.usatoday.com/story/entertainment/tv/2022/11/11/tulsa-king-review-sylvester-stallone-first-tv-show-mobster-mess/8323288001/ |url-status=live}}&lt;/ref&gt;

3395 | [[Sylvester Stallone]]

3396 | [[Sylvester Stallone]]

3397 | '''''[[Nanny (film)|Nanny]]''''' || [[Nikyatu Jusu]] || [[Anna Diop]], [[Michelle Monaghan]], [[Sinqua Walls]], [[Morgan Spector]], Rose Decker, [[Leslie Uggams]] || United States || Horror-thriller&lt;ref&gt;{{Cite web |title=Nanny |url=https://press.amazonstudios.com/us/en/original-movies/nanny |access-date=July 31, 2022 |website=[[Amazon Studios]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=D'Alessandro |first=Anthony |date=August 1, 2022 |title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan' &amp; More |url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/ |access-date=August 1, 2022 |website=Deadline Hollywood}}&lt;/ref&gt;

3398 :I tend to agree. Many things which appear &quot;random&quot; to one person may be perfectly meaningful to another. I used the example on the policy discussion page-what if I spelled out my first name in ASCII hex code? That would be meaningless gibberish to anyone who doesn't know what it is, but would make perfect sense to anyone who knows how to read it. What about foreign names? Do all admins know all foreign languages? Foreign words? The list of problems this policy could cause go on and on. Username policy should be applied as '''narrowly''' as possible, and only when the username in question is itself a form of vandalism (includes slurs/obscenities/advertising/etc.), or obviously confusing or bad faith (&quot;Sylvester Stallone (unless it can be verified that's really who it is!), an obvious attempt to impersonate another user, that type of thing), it should be allowed. If the user turns out to be a vandal, well, ban 'em for vandalism. As for &quot;hard to type&quot;-that's a silly argument, these newfangled computer thingies have included copy and paste functionality for years and years now, and I don't imagine that going away anytime soon. [[User:Seraphimblade|Seraphimblade]] 09:59, 9 November 2006 (UTC)

3399 According to the streaming aggregator Reelgood, ''Welcome to Wrexham'' was the 7th most watched program across all platforms, during the week of September 2, 2022.&lt;ref&gt;{{Cite web |last=Team |first=Mashable |date=September 3, 2022 |title=The most watched TV and movies of the week: Targaryens, twins, and therapists rule |url=https://mashable.com/article/most-watched-tv-shows-movies-streaming-what-to-watch-sept-2 |url-status=live |archive-url=https://web.archive.org/web/20221002205059/https://mashable.com/article/most-watched-tv-shows-movies-streaming-what-to-watch-sept-2 |archive-date=October 2, 2022 |access-date=October 4, 2022 |website=[[Mashable]]}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |date=September 2, 2022 |title=Reelgood: HBO Max's 'House of the Dragon' Tops Weekly Streaming; Sylvester Stallone's 'Samaritan' Has Strong Debut - Media Play News |url=https://www.mediaplaynews.com/reelgood-hbo-maxs-house-of-the-dragon-tops-weekly-streaming-sylvester-stallones-samaritan-has-strong-debut/ |url-status=live |archive-url=https://web.archive.org/web/20221004035542/https://www.mediaplaynews.com/reelgood-hbo-maxs-house-of-the-dragon-tops-weekly-streaming-sylvester-stallones-samaritan-has-strong-debut/ |archive-date=October 4, 2022 |access-date=October 4, 2022 |website=[[Media Play News]]}}&lt;/ref&gt;

3400 *[[:Sylvester Stallone]] (1 edit): [[Special:Diff/1038206874|(+439)]]

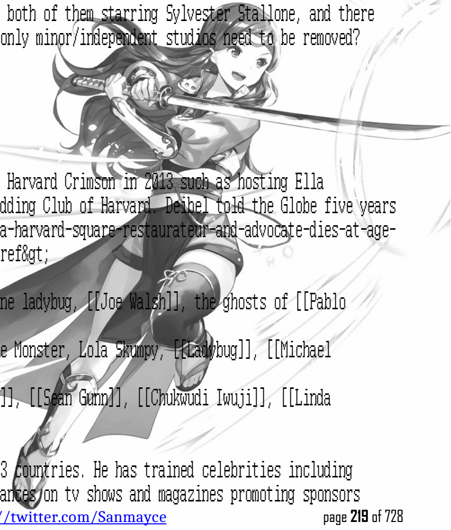
3401 *:::Rosguill, Although Samthong High School did not have brilliant teachers and students who made careers like Tiger Wood, Michael Jordan, and others such as Russell Wilson, Rob Gronkowski, Patrick Mahomes, J. J. Watt, Aaron Rodgers, Ben Roethlisberger and more in the columns of newspapers and books, but don't forget that there was one of the Principals of Samthong High School and one of the teachers who joined General Vang Pao's army to protect the American soldiers. All this deserves to be said and quoted here, especially in my article &quot;Samthong college&quot;. Do you have to have courage like these people to protect someone from elsewhere for good or for worse? it was courage that they did and defended the american soldiers. Not only Tou-Fu Vang, who was the Head Director, there was Wangyee Vang who was a teacher, then there was also Phon Yang. They put their lives on the line to protect others without realizing what is about to happen, so their careers and lives don't deserve to say &quot;notability&quot; value here in this article? However, there is Lormong Lo who has started his studies at Samthong College. It does not deserve to quote here? And you, during this time, where were you? With Silver Stalone to shoot the movie &quot;Rambo&quot;? [[Special:Contributions/2601:448:8400:7E80:0:0:0:170D|2601:448:8400:7E80:0:0:0:170D]] ([[User talk:2601:448:8400:7E80:0:0:0:170D|talk]]) 15:27, 6 July 2022 (UTC)

3402 * ''[[Chuck (film)|Chuck]]'' (2016) - biographical [[Sports film|sports]] drama film based on the life of heavyweight boxer [[Chuck Wepner]] and his [[Muhammad Ali vs. Chuck Wepner|1975 title fight]] with the heavyweight champion, [[Muhammad Ali]], which inspired [[Sylvester Stallone]]'s character and screenplay for the 1976 film ''[[Rocky]]''

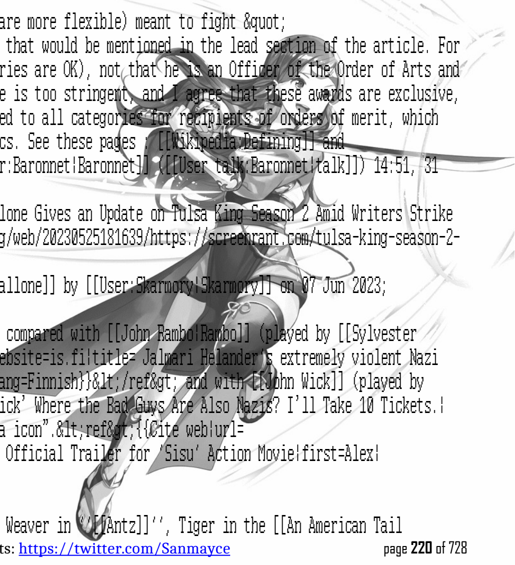
3403 In November 2021, [[Dolph Lundgren]] revealed that there are developments for a film centered around his character Ivan Drago.&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/movies/movie-features/dolph-lundgren-aquaman-2-drago-spinoff-1235053570/|work=The Hollywood Reporter|title=Dolph Lundgren on 'Castle Falls', 'Aquaman 2' and Drago Spinoff Talk|last=Davids|first=Brian|date=November 29, 2021|access-date=December 29, 2022}}&lt;/ref&gt; By July 2022, the project was officially announced by [[Metro-Goldwyn-Mayer|MGM]] with Robert Lawton serving as screenwriter on the project. Lawton had previously completed a [[spec script]] for a movie pitch centered around the making of the first ''Rocky'' film. Though the studio did not choose to develop that script, studio executives were impressed enough to hire the writer for the ''Drago'' film. The screenplay is stated to be centered around Ivan Drago's backstory.&lt;ref&gt;{{cite web|url=https://www.thewrap.com/creed-drago-spinoff-movie-robert-lawton-writer|work=The Wrap|title='Rocky' Spinoff Film 'Drago' Set for MGM With Robert Lawton to Write (Exclusive)|last=Gonzalez|first=Umberto|date=July 28, 2022|access-date=December 29, 2022}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url=https://variety.com/2022/film/news/rocky-franchise-drago-spinoff-1235328404/|work=Variety|title='Rocky' Film Franchise Expands With 'Drago' Spinoff|last=Rubin|first=Rebecca|date=July 28, 2022|access-date=December 29, 2022}}&lt;/ref&gt; The announcement of the spin-off movie, was met with criticism by franchise creator [[Sylvester Stallone]]. He stated that while he has tried to attain part of the rights to a franchise he was instrumental in creating, the producers have been &quot;exploiting&quot; his characters while trying to keep him out of the creative team.&lt;ref&gt;{{cite web | url=https://comicbook.com/movies/news/rocky-star-sylvester-stallone-slams-drago-spinoff-irwin-winkler-exploiting/?mibextid=zGKedc&amp;fs=e&amp;s=cl | title=Rocky Star Sylvester Stallone Slams Drago Spinoff}}&lt;/ref&gt; Following Stallone's response, Lundgren stated that he immediately reached out to his friend, explaining that when the project had been presented to him he had been informed that Stallone would be involved as a producer, as well as appearing in the movie; the actor further stated that the project isn't officially [[Green-light|green-lit]] yet.&lt;ref&gt;{{cite web|url=https://www.cinemablend.com/movies/rockys-dolph-lundgren-responds-after-sylvester-stallone-slams-the-developing-drago-movie|work=CinemaBlend|title=Rocky's Dolph Lundgren Responds After Sylvester Stallone Slams The Developing Drago Movie|date=August 1, 2022|access-date=December 29, 2022}}&lt;/ref&gt; While Stallone's reactions were perceived as uncharacteristic to his public image, he later stated that he has since discussed the potential movie with Lundgren.&lt;ref&gt;{{cite web|url=https://www.hollywoodreporter.com/tv/tv-features/sylvester-stallone-interview-rocky-rambo-tulsa-king-1235254384/|work=The Hollywood Reporter|title=Sylvester Stallone Gets Candid About Career, Regrets, Feuds: &quot;I Thought I Knew Everything&quot;|last=Hibberd|first=James|date=November 7, 2022|access-date=December 29, 2022}}&lt;/ref&gt;

3404 Reelgood, which tracks real-time streaming data from 5 million U.S. users, reported ''The Patient'' as the eighth most-watched show by August 31, 2022.&lt;ref&gt;{{Cite web |last=Gruenwedel |first=Erik |date=September 2, 2022 |title=Reelgood: HBO Max's 'House of the Dragon' Tops Weekly Streaming; Sylvester Stallone's 'Samaritan' Has Strong Debut |url=https://www.mediaplaynews.com/reelgood-hbo-maxs-house-of-the-dragon-tops-weekly-streaming-sylvester-stallones-samaritan-has-strong-debut/ |access-date=September 18, 2024 |website=[[Media Play News]] |language=en-US}}&lt;/ref&gt; By September 9, it had moved up to fourth place.&lt;ref&gt;{{Cite web |date=September 9, 2022 |title=The most watched TV and movies of the week: 'The Rings of Power' rules them all |url=https://mashable.com/article/most-watched-tv-shows-movies-streaming-what-to-watch-sept-9 |url-status=live

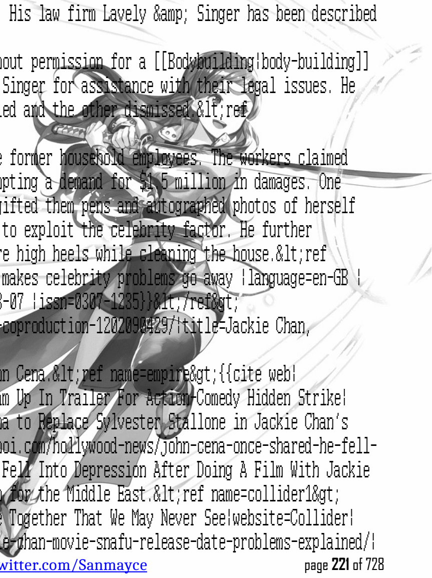
```
|archive-url=https://web.archive.org/web/20221002233453/https://mashable.com/article/most-watched-tv-shows-movies-streaming-what-to-watch-sept-9 |archive-date=October 2, 2022 |access-date=October 4, 2022 |website=[[Mashable]]}}&lt;/ref&gt; Additionally, [[JustWatch]], which collects data from over 20 million users globally, ranked the series as the third most-streamed in the U.S. between September 5-12.&lt;ref&gt;{{Cite web |last=Gruenwedel |first=Erik |date=September 12, 2022 |title=JustWatch: 'Top Gun: Maverick,' 'Rick and Morty' Top Weekly Streaming Through Sept. 11 |url=https://www.mediaplaynews.com/justwatch-top-gun-maverick-rick-and-morty-top-weekly-streaming-through-sept-11/ |access-date=September 18, 2024 |website=[[Media Play News]] |language=en-US}}&lt;/ref&gt;
3405  | ShortSummary    = After Terry painted the ceiling, Korvo attempts to turn the rest of the Solar Opposites into gargoyles to give him some alone time. His gun misfires into a neighbor's house, and the police are called. Assuming that Terry and Korvo are the children in the family, they are taken away by Child Protective Services. Korvo ends up on a bus in the tunnel used to film the movie [[Daylight_(1996_film)|Daylight]] when the bus breaks down and the tunnel begins collapsing. [[Sylvester Stallone]], who starred in the movie, is also on the bus, and helps rescue Korvo and the other passengers. Terry is believed to have special abilities and is set to be studied by the US government, but he is rescued by a scientist who is surprised to find that Terry is an adult. Yumyulack and Jesse enter the mind of their school principal and realize that he is repressing anger over being forced to be a principal despite loving playing [[trumpet]].  They get him to embrace trumpet playing, and escape from his mind.
3406 Pro wrestler [[PB Smooth]] and former [[Sylvester Stallone]] bodyguard [[Matt Hannon|Matt Karedas]] are also attached to the film.&lt;ref name=&quot;pwmania&quot; /&gt;&lt;ref name=&quot;bands&quot; /&gt;
3407 Ian Sedensky of ''Culture Crypt'' gave the film a score of 20 out of 100, writing that &quot;meanders well past the point of fleshing out everyone with backstories and over the border into full-on &quot;where's the shark already?&quot; boredom.&quot;&lt;ref&gt;{{cite web |url= https://culturecrypt.com/movie-reviews/maneater-2022|title= MANEATER (2022)|last= Sedesnky|first= Ian|date= 29 August 2022|website= Culture Crypt|publisher= |access-date= 14 September 2022|quote=}}&lt;/ref&gt; Matthew Monagle of ''[[The Austin Chronicle]]'' rated the film 0.5 stars out of 5, calling it a &quot;particularly rough watch&quot;.&lt;ref&gt;{{cite web |url= https://www.austinchronicle.com/events/film/2022-08-26/maneater/|title= Maneater|last= Monagle|first= Matthew|date= 26 August 2022|website= [[The Austin Chronicle]]|publisher= |access-date= 14 September 2022|quote=}}&lt;/ref&gt; Noel Murray of the ''[[Los Angeles Times]]'' wrote that &quot;By the end, &quot;Maneater&quot; has walked right up to the edge of being a fun, silly, &quot;so bad it's good&quot; time-killer. But after taking way too long, it never really arrives there.&quot;&lt;ref&gt;{{cite web |url= https://www.latimes.com/entertainment-arts/movies/story/2022-08-26/samaritan-amazon-stallone-me-time-running-with-devil-mcafee-netflix|title= Review: Sylvester Stallone finds nuance in superhero deconstruction 'Samaritan'|last= Murray|first= Noel|date= 26 August 2022|website= [[Los Angeles Times]]|publisher= |access-date= 14 September 2022|quote=}}&lt;/ref&gt;
3408 | {{flagicon|USA}} [[Sylvester Stallone]]
3409 | {{flagicon|USA}} [[Sylvester Stallone]]
3410 | {{flagicon|USA}} [[Sylvester Stallone]]
3411 | {{flagicon|USA}} [[Sylvester Stallone]]
3412 The film was released on [[Prime Video]] on December 9, 2022.&lt;ref name=&quot;deadline1&quot;&gt;{{cite news |last1=D'Alessandro |first1=Anthony |title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan', 'Good Night Oppy' &amp; More |url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/ |access-date=29 August 2022 |publisher=Deadline |date=1 August 2022 |archive-date=October 15, 2022 |archive-url=https://web.archive.org/web/20221015083954/https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/ |url-status=live }}&lt;/ref&gt;
3413 '''''Running With the Devil: The Wild World of John McAfee''''' is a 2022 documentary film about [[John McAfee]].&lt;ref&gt;{{Cite web|url=https://www.cnn.com/2022/08/24/entertainment/running-with-the-devil-review/index.html|title='Running With the Devil' can't make sense of John McAfee's wild world|website=CNN}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.latimes.com/entertainment-arts/movies/story/2022-08-26/samaritan-amazon-stallone-me-time-running-with-devil-mcafee-netflix|title=Review: Sylvester Stallone finds nuance in superhero deconstruction 'Samaritan'|first1=Noel Murray|last1=Aug. 26|first2=2022 6 Am|last2=Pt|date=August 26, 2022|website=Los Angeles Times}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.gq-magazine.co.uk/culture/article/running-with-the-devil-netflix-documentary-interview|title=The wild story behind The Wild World of John McAfee|website=British GQ}}&lt;/ref&gt;
3414 Nagler practices law in Los Angeles, California.&lt;ref name=&quot;auto3&quot;/&gt; He has represented, among others, tennis players Arthur Ashe and [[Tom Okker]], and actor [[Sylvester Stallone]].&lt;ref&gt;{{Cite web|url=https://www.hollywoodreporter.com/business/business-news/pellicanos-clients-deserved-a-more-111042/|title=Pellicano's clients deserved a more thorough cross-examination|work=Hollywood Reporter|author=Matthew Belloni|date=May 6, 2008}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.nytimes.com/1968/10/20/archives/gift-of-100-gm-shares-worth-8900-keeps-arthur-ashe-smiling.html|author=Dave Anderson|title=Gift of 100 GM Shares Worth $8,900 Keeps Arthur Ashe Smiling!|date=October 20, 1968|website=The New York Times}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://www.youtube.com/watch?v=tKL7pvE9MbU|author=Larry Nagler|title=Life lessons learned on the tennis court. &amp;#124; TEDxBoston (''video'')|via=YouTube|date=April 7, 2022}}&lt;/ref&gt; He was a Judge Pro Tem of the [[Beverly Hills]] Municipal Court and Los Angeles Municipal Court in 1976-78.&lt;ref name=&quot;auto1&quot;/&gt;&lt;ref&gt;{{cite book | url=https://books.google.com/books?id=TwE8AQAAIAAJ&amp;q=larry+nagler+Beverly+Hills+Municipal+Court | title=The Martindale-Hubbell Law Directory | date=February 5, 1998 | publisher=Martindale-Hubbell Law Directory, Incorporated }}&lt;/ref&gt;
3415 *Sylvester Stallone should be linked in his caption.
3416 According to market research company Parrot Analytics, which looks at consumer engagement in consumer research, streaming, downloads, and on social media, ''She-Hulk: Attorney at Law'' experienced a strong second week, with a demand of 40.6 times the average series in its first full week since premiering.&lt;ref&gt;{{Cite web |date=August 31, 2022 |title=Breakout Shows (August 20 - 26, 2022): House of the Dragon soars to the top of the ranking in its first week |url=https://www.parrotanalytics.com/insights/breakout-shows-august-20-26-house-of-the-dragon-soars-to-the-top-of-the-ranking-in-its-first-week/ |access-date=2024-09-27 |website=Parrot Analytics |language=en}}&lt;/ref&gt;&lt;ref name=&quot;ParrotAnalytics&quot; /&gt; [[Nielsen Media Research]], which records streaming viewership on U.S. television screens, calculated that it was the ninth-most watched original series across streaming services for the week of August 22 to 28, 2022 with 390 million minutes watched.&lt;ref name=&quot;MinutesWatched&quot; /&gt; The streaming aggregator Reelgood, which monitors real-time data from 5 million users in the U.S. for original and acquired streaming programs and movies across subscription video-on-demand (SVOD) and ad-supported video-on-demand (AVOD) services, reported that ''She-Hulk: Attorney at Law'' was the second most-streamed series for the week ending August 27, 2022,&lt;ref name=&quot;ReelGood&quot; /&gt; and was the fourth most-streamed program overall for the week ending August 31, 2022.&lt;ref&gt;{{Cite web |last= |first= |date=September 3, 2022 |title=The most watched TV and movies of the week: Targaryens, twins, and therapists rule |url=https://mashable.com/article/most-watched-tv-shows-movies-streaming-what-to-watch-sept-2 |access-date=2024-09-27 |website=[[Mashable]] |language=en}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Gruenwedel |first=Erik |date=September 2, 2022 |title=Reelgood: HBO Max's 'House of the Dragon' Tops Weekly Streaming; Sylvester Stallone's 'Samaritan' Has Strong Debut |url=https://www.mediaplaynews.com/reelgood-hbo-maxs-house-of-the-dragon-tops-weekly-streaming-sylvester-stallones-samaritan-has-strong-debut/?hilite=She-Hulk |access-date=2024-09-27 |website=[[Media Play News]] |language=en-US}}&lt;/ref&gt; [[Whip Media]], which tracks viewership data for the more than 21 million worldwide users of its [[TV Time]] app, calculated that it was the second most-streamed original series for the week ending August 28, 2022.&lt;ref&gt;{{Cite web |last=Prange |first=Stephanie |date=August 30, 2022 |title='Me Time,' 'Only Murders in the Building' Top Weekly Whip U.S. Streaming Charts |url=https://www.mediaplaynews.com/me-time-only-murders-in-the-building-top-weekly-whip-u-s-streaming-charts/?hilite=She-Hulk |access-date=2024-09-27 |website=[[Media Play News]] |language=en-US}}&lt;/ref&gt;
3417 &lt;ref name=&quot;ReelGood&quot;&gt;{{Cite web |date=August 27, 2022 |title=The most watched TV and movies of the week: 'House of the Dragon' soars to the top |url=https://mashable.com/article/most-watched-tv-shows-movies-streaming-what-to-watch-aug-27 |url-status=live |archive-url=https://web.archive.org/web/20220926055721/https://mashable.com/article/most-watched-tv-shows-movies-streaming-what-to-watch-aug-27 |archive-date=September 26, 2022 |access-date=October 4, 2022 |website=[[Mashable]]}}&lt;/ref&gt;&lt;!-- Additional source that uses a different date range and has a different rank: {{Cite web |last=Gruenwedel |first=Eric |date=September 2, 2022 |title=Reelgood: HBO Max's 'House of the Dragon' Tops Weekly Streaming; Sylvester Stallone's 'Samaritan' Has Strong Debut |url=https://www.mediaplaynews.com/reelgood-hbo-maxs-house-of-the-dragon-tops-weekly-streaming-sylvester-stallones-samaritan-has-strong-debut/ |url-status=live |archive-url=https://web.archive.org/web/20220902161253/https://www.mediaplaynews.com/reelgood-hbo-maxs-house-of-the-dragon-tops-weekly-streaming-sylvester-stallones-samaritan-has-strong-debut/ |archive-date=September 2, 2022 |access-date=October 4, 2022 |website=[[Media Play News]]}} --&gt;
3418 ::::I said ''&quot;either '''mostly''' or '''only''' their own films&quot;'', which is exactly what those categories represent. ''Syncopy Inc. films'' only has two films that weren't directed by Nolan, ''Infinitum Nihil films'' only has one film that Depp hasn't worked in, and the same goes for the other categories I listed, most of the films were either directed by or starring the person who owns the company, they all started out by producing their own films. And there's also {{lc|Balboa Productions films}}, with only 2 films and both of them starring Sylvester Stallone, and there are probably other categories in the same situation. Either we get rid of all of them and create a new rule or leave it as it is right now. Why only minor/independent studios need to be removed? [[User:Zoolver|Zoolver]] ([[User talk:Zoolver|talk]]) 04:56, 10 October 2022 (UTC)
3419 * [[Sylvester Stallone]] - ''[[Rhinestone (film)|Rhinestone]]'' (1984)
3420 * [[Sylvester Stallone]] - ''[[Rambo: First Blood Part II]]'' and ''[[Rocky IV]]'' (1985)
3421 * [[Sylvester Stallone]] - ''[[Rambo III]]'' (1988)
3422 * [[Sylvester Stallone]] - ''[[Stop! Or My Mom Will Shoot]]'' (1992)
3423 Deibel handled the front of the restaurant, greeting guests warmly, while Hughes was the executive chef. The pair reflected on memories with the Harvard Crimson in 2013 such as hosting Ella Fitzgerald, Clint Eastwood, Sylvester Stallone and Robin Williams as well as countless other Man and Woman of the Year nominated by the Hasty Pudding Club of Harvard. Deibel told the Globe five years ago that her favorite customer was Madeleine Albright.&lt;ref&gt;{{cite news |url=https://www.cambridgeday.com/2023/06/06/mary-catherine-deibel-a-harvard-square-restaurateur-and-advocate-dies-at-age-72/ |title=Mary-Catherine Deibel, a Harvard Square restaurateur and advocate, dies at age 72 |publisher=Cambridge Day | date=June 6, 2023}}&lt;/ref&gt;
3424 * [[List of awards and nominations received by Sylvester Stallone|Sylvester Stallone]]
3425 * [[Nick Kroll]] as Nick Birch, [[Maurice the Hormone Monster]], Coach Steve, Rick the Hormone Monster, Lola, the [[Statue of Liberty]], a profane ladybug, [[Joe Walsh]], the ghosts of [[Pablo Picasso|Picasso]] and [[Richard Burton]], [[Sylvester Stallone]] and a webcam girl
3426 * [[Nick Kroll]] as Nick Birch, [[Maurice the Hormone Monster]], Coach Steve Steve ([[Theodore Roosevelt|President Roosevelt]]), Rick the Hormone Monster, Lola Skumpy, [[Ladybug]], [[Michael Shannon]], [[Raccoons]], [[Vin Diesel]], [[Joe Brown (judge)|Judge Joe Brown]], Mila and Lotte Jansen, [[Sylvester Stallone]] and others
3427 | [[Chris Pratt]], [[Zoe Saldaña]], [[Dave Bautista]], [[Karen Gillan]], [[Pom Klementieff]], [[Vin Diesel]], [[Bradley Cooper]], [[Will Poulter]], [[Sean Gunn]], [[Chukwudi Iwuji]], [[Linda Cardellini]], [[Nathan Fillion]] and [[Sylvester Stallone]]
3428 | [[Sylvester Stallone]]
3429 * [[Alex Morgan motonautica]] - (World Cup and European Cup Jet Ski Champion, Alex participated in more than 50 international racing events in 13 countries. He has trained celebrities including Cristiano Ronaldo, Alejandro Fernandes, and Fher from Mana. In Mexico he received the award Water Craft Racer of the Century. He has made appearances on tv shows and magazines promoting sponsors
```



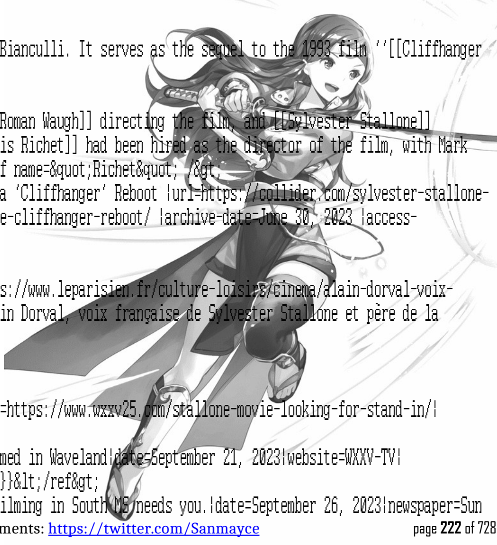
```
including: Trump, Don Julio, Bomba, Riva, Sea Doo, Yamaha, Gran Orendain. Alex Morgan has also been selected President of the jury at Miss Italia Beauty Pageant along with celebrities Bruce Willis, Sylvester Stallone and Eros Ramazzotti. Morgan participated on trials for the Maserati Racing Team in Mugello Italy and qualified to take part in the Maserati corsa world series. In 2015, Morgan was named Ambassador for the state of Jalisco at the Milan, Italy World Expo, helping to promote trade with Mexico in Italy and worldwide)  (http://www.p1aquax.com/news/AQX91357.ink https://www.youtube.com/watch?v=osIfGrBvx98 http://www.news-journalonline.com/article/20130613/sports/130619869 http://mephim.net/xem-phim/Morgan-jet-ski-world-tour https://www.youtube.com/watch?v=gWiootasZXs https://www.youtube.com/watch?v=Kj88RUXP0Vo http://www.milenio.com/region/Campeon-mundial-motonautica-competira-Florida_0_400160346.html http://domino.fimconi.it/query_calendari_fim.nsf/AG05?OpenAgent&amp;IDP=324CA0E06B34D844C1256BCF00330195 https://twitter.com/veletti/status/720790875970736128 http://forwom.net/video/healthy-happy/entrevista-con-alex-morgan-campeo-n-mundial-de-motona-utica.html http://www.nautica.it/notizie/nauticaonline/2003/01/07/230306/motonautica-copa-de-america-endurance-jisba http://www.elmundo.es/elmundomotor/2001/05/18/ocio/990208570.html https://www.facebook.com/MexicoChallenge/ http://www.losportitaliano.it/2002/news/mag02/motonautica12.htm http://www.ytmp3dl.com/Entrevista-con-Alex-Morgan-Campeo%CC%81n-Mundial-de-Motona%CC%81utica(Kj88RUXP0Vo) http://surf-and-snow.info/uim_report2.htm http://www.deporte.org.mx/pag/noticias/noticias_leer.asp?id=989 http://mural-guadalajara.vlex.com.mx/vid/busca-morgan-vergara-79373663 http://mural-guadalajara.vlex.com.mx/vid/huele-morgan-victoria-79401401 https://www.youtube.com/watch?v=HTDWDNZ9jQc)
3430 * [[Sylvester Stallone]] - ''[[Spy Kids 3-D: Game Over]]'' (2003)
3431 * [[Sylvester Stallone]] - ''[[Expend4bles]]'' (2023)
3432 * [[Sylvester Stallone]] - ''[[Rocky IV]]'' (1985)
3433 ** [[Sylvester Stallone]] - ''[[Samaritan (film)|Samaritan]]'' as Joe Smith / Samaritan / Nemesis
3434 During this period, he successfully defended his AWF title against [[National Wrestling Alliance]] star [[Nikita Koloff]].&lt;ref&gt;{{cite news |author=&lt;!--Staff writer(s); no by-line.--&gt; |title=AWF schedules show at York |work=Hanover Evening Sun |location=Hanover, Pennsylvania |date=May 24, 1989 |page=23 |access-date= }}&lt;/ref&gt; as well as ex-[[World Wide Wrestling Federation]] veterans [[Afa the Wild Samoan]]&lt;ref&gt;{{cite news |author=&lt;!--Staff writer(s)/no by-line.--&gt; |date=August 4, 1989 |title=Richter, Bounty Hunter keep titles |page=14 |work=Hanover Evening Sun |location= |access-date=}}&lt;/ref&gt; and [[Chief Jules Strongbow]].&lt;ref name=&quot;Monsour&quot;/&gt; He also made appearances for other local Mid-Atlantic promotions including the [[Neil Superior|National Wrestling League]] and [[World Xtreme Wrestling|Trans World Wrestling Federation]]. In June 1990, Thrasher went on his first international tour with the NWL to Guam. It was on this tour that he lost the AWF Heavyweight Championship to John Rambo at the [[University of Guam]] after a two-year title reign.&lt;ref name=&quot;WTH-AWF1&quot;/&gt; The show was seen by 3,600 fans and included Gov. [[Joseph F. Ada]] and Congressional Del. [[Ben Blaz]] in attendance.&lt;ref&gt;{{cite news |last=Mayer |first=Joe |date=June 22, 1990 |title=Super Slam '90 draws 3,600 |page= |work=[[Pacific Daily News]] |location= |access-date=}}&lt;/ref&gt; Their match was featured on the front page of the ''[[Pacific Daily News]]''.&lt;ref&gt;{{cite news |last=Roller |first=Bob |date=June 23, 1990 |title=GRAPPLING ACTION |page=1 |work=[[Pacific Daily News]] |location= |access-date=}}&lt;/ref&gt; Their rematch the following night was seen by 4,500 people. After being pinned by Rambo once again, Thrasher attacked referee Dave &quot;The Wave&quot; McCallister and was fined $1,000.&lt;ref&gt;{{cite news |last=Mayer |first=Joe |date=June 24, 1990 |title=Super Slam wrap-up; Wrestlers draw 4500 enthusiastic fans |page=10 |work=[[Pacific Daily News]] |location= |access-date=}}&lt;/ref&gt; Thrasher has attributed the show's popularity due to foreign wrestling fans believing [[Sylvester Stallone]] was actually portraying Rambo.&lt;ref name=&quot;PIN-INT1&quot;/&gt;
3435 * [[Sylvester Stallone]]
3436 * {{icon|B}} [[:Sylvester Stallone]]
3437 Beven conducted his first solo exhibition in 1958 at the [[Royal Empire Society]].&lt;ref name=&quot;:2&quot;&gt;{{Cite web |title=Remembering George Beven - the man and the artist |url=https://www.sundaytimes.lk/230212/plus/remembering-george-beven-the-man-and-the-artist-511083.html |access-date=18 February 2023 |website=Print Edition - The Sunday Times, Sri Lanka}}&lt;/ref&gt; He also made a name for himself with his ability of painting in oils using his fingers. He also cemented his reputation and stamped his authority in tooth brush paintings during the 1970s, 1980s and 1990s, which portrayed many famous artists of the time including the likes of [[Margot Fonteyn]], [[Rudolf Nureyev]], [[Mikhail Baryshnikov|Mikhail Baryshmikov]] from the [[classical ballet]] as well as veteran Hollywood actors such as [[Marlene Dietrich]], [[Judy Garland]], [[Liza Minnelli]], [[Marilyn Monroe]], [[James Dean]] and [[Sylvester Stallone]].&lt;ref name=&quot;:0&quot; /&gt; He was also a clever artist having done lot of research and experimenting with poster colours and gouache using the most unlikely colour combinations in order to provoke the eyes of the viewers.
3438 ** [[Sylvester Stallone]] - ''[[Tulsa King]]'' (Paramount+) as Dwight Manfredi
3439 Predator eradication attempts began in 2019, shortly after the predator-proof fencing was completed.&lt;ref name=&quot;:4&quot; /&gt; It took about a year for conservation managers to remove six feral cats and five red foxes from the pest-free zone; however, a sixth fox proved more elusive.&lt;ref name=&quot;:2&quot; /&gt; This fox, likely a [[Red fox#Terminology|kit]] born in 2018, is believed to have lost his mother to a trap and his sibling to [[Sodium fluoroacetate|1080-bait]], which may have made him more wary than a typical fox.&lt;ref name=&quot;:4&quot;&gt;{{Cite web |last=Fontaine |first=Angus |date=20 May 2022 |title=Baiting. Shoot nights. Sniffer dogs. 4500 man hours: The fox that won't be caught |url=https://www.smh.com.au/national/baiting-shoot-nights-sniffer-dogs-4500-man-hours-the-fox-that-won-t-be-caught-20220427-p5agm1.html |url-status=live |archive-url=https://web.archive.org/web/20230315175635/https://www.smh.com.au/national/baiting-shoot-nights-sniffer-dogs-4500-man-hours-the-fox-that-won-t-be-caught-20220427-p5agm1.html |archive-date=15 March 2023 |access-date=15 March 2023 |website=[[The Sydney Morning Herald]] |language=en}}&lt;/ref&gt; Although photographed by infrared [[camera trap]]s about every three months, the fox was rarely seen in person.&lt;ref name=&quot;:2&quot; /&gt; After three years, he had only twice been seen by humans: the first time in 2019, when he briefly entered a staff camp, and again in 2020, when he was flushed out by [[Scent hound|scent-tracking dogs]].&lt;ref name=&quot;:2&quot; /&gt; Given his proclivity for avoiding capture, the fox was nicknamed Rambo, after the [[John Rambo|character]] played by [[Sylvester Stallone]] in the 1982 film ''[[First Blood]]''.&lt;ref name=&quot;:1&quot; /&gt;
3440 Actor [[Monsour del Rosario]] stated that the film's story is inspired by the 1995 film ''[[Assassins (1995 film)|Assassins]]'' starring [[Sylvester Stallone]] and [[Antonio Banderas]].&lt;ref&gt;{{cite news|last=Lo|first=Ricky|title=Why Monsour said no to Probinsyano not once but twice|url=https://www.philstar.com/entertainment/2018/09/22/1853464/why-monsour-said-no-probinsyano-not-once-twice|access-date=February 16, 2023|work=[[The Philippine Star|Philstar.com]]|publisher=Philstar Global Corp.|date=September 22, 2018}}&lt;/ref&gt;
3441 | [[Sylvester Stallone]], LIVE's Bread Club Week
3442 | '''LIVE's Virtual Road Trip - [[Cleveland, Ohio]]''': [[Sylvester Stallone]], [[Judd Hirsch]], Chef [[Michael Symon]]
3443 :The result of the discussion was &lt;/noinclude&gt;'''retarget'''&lt;includeonly&gt;&lt;/div&gt;&lt;/includeonly&gt;&lt;noinclude&gt;&lt;!-- Template:Rfd top--&gt; to [[Sylvester Stallone#Early life]]. &lt;small&gt;[[Wikipedia:NACD|(non-admin closure)]]&lt;/small&gt; -[[User:LaundryPizza03|&lt;b style=&quot;color:#77b&quot;&gt;Laundry&lt;/b&gt;&lt;b style=&quot;color:#fb0&quot;&gt;Pizza&lt;/b&gt;&lt;b style=&quot;color:#b00&quot;&gt;03&lt;/b&gt;]] ([[User talk:LaundryPizza03|&lt;span style=&quot;color:#0d0&quot;&gt;d&lt;/span&gt;]][[Special:Contribs/LaundryPizza03|&lt;span style=&quot;color:#0bf&quot;&gt;c̄&lt;/span&gt;]]) 07:40, 2 March 2023 (UTC)
3444 *&lt;span id=&quot;Michael Sylvester Gardenzio Stallone&quot;&gt;{{no redirect|1 = Michael Sylvester Gardenzio Stallone }}&lt;/span&gt; → [[:Sylvester Stallone]] &lt;span&gt;&amp;nbsp;&lt;span class=&quot;plainlinks lx&quot;&gt;([[Talk:Michael Sylvester Gardenzio Stallone|talk]] &lt;b&gt;·&lt;/b&gt; [[Special:WhatLinksHere/Michael Sylvester Gardenzio Stallone|links]] &lt;b&gt;·&lt;/b&gt; [[Special:PageHistory/Michael Sylvester Gardenzio Stallone|history]] &lt;b&gt;·&lt;/b&gt; [[:toolforge:pageviews/?start=2023-01-22&amp;end=2023-02-20&amp;project=en.wikipedia.org&amp;pages=Michael_Sylvester_Gardenzio_Stallone|stats]])&lt;/span&gt;&lt;/span&gt;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&lt;small class=&quot;plainlinks&quot;&gt;&lt;nowiki&gt;[&lt;/nowiki&gt;&amp;nbsp;Closure:&amp;nbsp;''{{#ifeq:{{FULLPAGENAME}}|Wikipedia:Redirects for discussion|(@subpage)|[{{fullurl:Michael Sylvester Gardenzio Stallone|action=edit&amp;summary={{Urlencode:[[{{FULLPAGENAME}}#Michael Sylvester Gardenzio Stallone]] closed as keep}}}} keep]/[{{fullurl:Michael Sylvester Gardenzio Stallone|action=edit&amp;summary={{Urlencode:[[{{FULLPAGENAME}}#Michael Sylvester Gardenzio Stallone]] closed as retarget}}}} retarget]&lt;span class=&quot;sysop-show&quot;&gt;/[{{fullurl:Michael Sylvester Gardenzio Stallone|action=delete&amp;wpReason={{Urlencode:[[{{FULLPAGENAME}}#Michael Sylvester Gardenzio Stallone]] closed as delete}}&amp;wpMovetalk=1}} delete]&lt;/span&gt;}}''&amp;nbsp;]&lt;/small&gt;&amp;nbsp;
3445 *'''Weak refine''' to [[Sylvester Stallone#Early life]] where this is explained but keeping as is per CycloneYoris is also fine. [[User:A7V2|A7V2]] ([[User talk:A7V2|talk]]) 23:51, 21 February 2023 (UTC)
3446 *:&quot;Defining&quot;, according to this very debatable criterion of the English Wikipedia only (other language-versions of Wikipedia are more flexible) meant to fight &quot;[[:Wikipedia:Overcategorization|overcategorization]]&quot; (i.e. too many categories for each article), signifies essentially a quality that would be mentioned in the lead section of the article. For example : the lead section of the article about Sylvester Stallone mentions that he is an American actor and filmmaker (so these categories are OK), not that he is an Officer of the Order of Arts and Letters (so although he indeed received this honour, this category is not OK, according to the English Wikipedia rule). I feel this rule is too stringent, and I agree that these awards are exclusive, but if this rule is applied stringently to delete &quot;Category:Recipients of the Order Pro Merito Melitensi&quot;, it should be applied to all categories for recipients of orders of merit, which are never the main defining characteristic for the persons who receive them. It is rather a consequence of other defining characteristics. See these pages : [[Wikipedia:Defining]] and [[Wikipedia:Overcategorization]]. I have not been able to find when and where this rule was created, and where it can be debated. [[User:Baronnet|Baronnet]] ([[User talk:Baronnet|talk]]) 14:51, 31 March 2023 (UTC)
3447 | &lt;ref&gt;{{Cite web |url=https://screenrant.com/tulsa-king-season-2-update-writers-strike-sylvester-stallone/ |title=Sylvester Stallone Gives an Update on Tulsa King Season 2 Amid Writers Strike |website=[[Screen Rant]] |date=May 18, 2023 |access-date=August 12, 2023 |archive-date=May 25, 2023 |archive-url=https://web.archive.org/web/20230525181639/https://screenrant.com/tulsa-king-season-2-update-writers-strike-sylvester-stallone/ |url-status=live }}&lt;/ref&gt;
3448 * 31 May 2023 &amp;ndash; [[Stallone]] move request to [[Stallone (surname)]] by [[User:BD2412|BD2412]] was ''moved'' to [[Sylvester Stallone]] by [[User:Skarmory|Skarmory]] on 07 Jun 2023; [[Talk:Stallone#Requested move 31 May 2023|discussion]]
3449 Tommila's character in the film ''[[Sisu (film)|Sisu]]'', filmed in [[Lapland (Finland)|Lapland]] with a budget of €6 million, has been compared with [[John Rambo|Rambo]] (played by [[Sylvester Stallone]]) in the [[First Blood (film)|1982 film]],&lt;ref&gt;{{Cite web|url=https://www.is.fi/tv-ja-elokuvat/art-2000009349834.html|website=is.fi|title= Jalmari Helander's extremely violent Nazi lynching film caused an extraordinary reaction in Hollywood|first=Taneli|last=Topelus|date=January 29, 2023|accessdate=April 21, 2023|lang=Finnish}}&lt;/ref&gt; and with [[John Wick]] (played by [[Keanu Reeves]]).&lt;ref&gt;{{cite web|url=https://www.themarysue.com/sisu-trailer-like-a-a-finnish-john-wick/|title=A Finnish 'John Wick' Where the Bad Guys Are Also Nazis? I'll Take 10 Tickets.|first=Kate|last=Hudson|publisher=The Mary Sue|access-date=22 April 2023}}&lt;/ref&gt; He has also been described as a "new action cinema icon".&lt;ref&gt;{{Cite web|url=https://www.firstshowing.net/2023/official-trailer-for-sisu-action-movie|website=Firstshowing.net|title=Meet Finland's New Action Hero: Official Trailer for 'Sisu' Action Movie|first=Alex|last=Billington|date=February 21, 2023|accessdate=April 21, 2023}}&lt;/ref&gt;
3450 | [[Sylvester Stallone]] &amp; Family, LIVE's I Love Mom Week - Laurie &amp; Misha Gelman
3451 * '''February 13''': [[Alain Dorval]], French voice actor (French dub voice of Pete in the [[Mickey Mouse|''Mickey Mouse'' franchise]], Weaver in ''[[Antz]]'', Tiger in the [[An American Tail
```



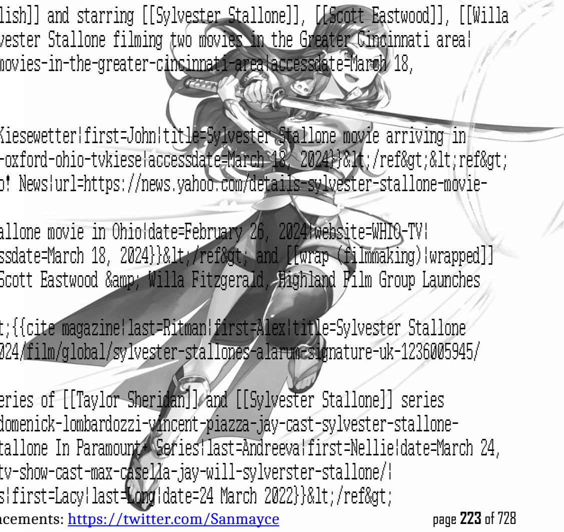
```
(franchise)|''American Tail'' franchise]], Victor in ''[[Ratchet &amp; Clank (film)|Ratchet &amp; Clank]]'', Lex Luthor in ''[[Superman: The Animated Series]]'', Sylvester Stallone in ''[[Solar Opposites]]'', voice of Goliath the Rhino in ''[[The Jungle Bunch (film)|The Jungle Bunch]]''), dies at age 77.&lt;ref&gt;{{Cite web |date=2024-02-13 |title=Mort du comédien Alain Dorval, la voix française de Sylvester Stallone |url=https://www.francetvinfo.fr/culture/cinema/mort-du-comedien-alain-dorval-la-voix-francaise-de-sylvester-stallone_6363346.html |access-date=2024-02-13 |website=Franceinfo |language=fr-FR |archive-date=2024-02-13 |archive-url=https://web.archive.org/web/20240213143505/https://www.francetvinfo.fr/culture/cinema/mort-du-comedien-alain-dorval-la-voix-francaise-de-sylvester-stallone_6363346.html |url-status=live }}&lt;/ref&gt;&lt;ref&gt;[https://www.libramemoria.com/deces-celebres/2024/02/13/le-doubleur-francais-alain-dorval-est-decede Le doubleur français Alain Dorval est décédé] {{Webarchive|url=https://web.archive.org/web/20240213145944/https://www.libramemoria.com/deces-celebres/2024/02/13/le-doubleur-francais-alain-dorval-est-decede |date=2024-02-13 }} {{in lang|fr}}&lt;/ref&gt;&lt;ref&gt;[https://www.leparisien.fr/culture-loisirs/cinema/alain-dorval-voix-francaise-de-sylvester-stallone-et-pere-de-la-ministre-aurore-berge-est-mort-a-77-ans-13-02-2024-LBRHLKLMV5EZXE4PCFFCEHLJYY.php Alain Dorval, voix française de Sylvester Stallone et père de la ministre Aurore Bergé, est mort à 77 ans] {{Webarchive|url=https://web.archive.org/web/20240213162319/https://www.leparisien.fr/culture-loisirs/cinema/alain-dorval-voix-francaise-de-sylvester-stallone-et-pere-de-la-ministre-aurore-berge-est-mort-a-77-ans-13-02-2024-LBRHLKLMV5EZXE4PCFFCEHLJYY.php |date=2024-02-13 }} {{in lang|fr}}&lt;/ref&gt;
 | Guests          = [[Sylvester Stallone]], Nicola Peltz Beckham, [[Gloria Gaynor]]
The festival's opener was the Japanese animated film ''[[The Boy and the Heron]]'' by [[Hayao Miyazaki]], which marked the first time in history that an animated film opened the festival.&lt;ref&gt;[https://www.thestar.com/entertainment/animation-master-hayao-miyazaki-s-the-boy-and-the-heron-to-open-tiff-this-year/article_88ffd818-f2b0-531c-a4a9-4ae3cdb45828.html &quot;Animation master Hayao Miyazaki's 'The Boy and the Heron' to open TIFF this year&quot;]. ''[[Toronto Star]]'', July 27, 2023.&lt;/ref&gt;&lt;ref&gt;Rebecca Ford, [https://www.vanityfair.com/hollywood/2023/07/hayao-miyazaki-the-boy-and-the-heron-open-the-toronto-film-festival-awards-insider &quot;Hayao Miyazaki's The Boy and the Heron Will Open the Toronto Film Festival&quot;]. ''[[Vanity Fair (magazine)|Vanity Fair]]'', July 27, 2023.&lt;/ref&gt; It concluded with ''[[Sly (film)|Sly]]'', [[Thom Zimny]]'s documentary film about actor [[Sylvester Stallone]];&lt;ref&gt;[https://www.theglobeandmail.com/arts/film/tiff/article-netflixs-sylvester-stallone-documentary-sly-to-close-tiff/ &quot;Netflix's Sylvester Stallone documentary Sly to close TIFF&quot;]. ''[[The Globe and Mail]]'', August 8, 2023.&lt;/ref&gt; Stallone also participated in an ''In Conversation'' event in conjunction with the film.&lt;ref&gt;Melanie Goodfellow, [https://deadline.com/2023/08/sylvester-stallone-career-at-toronto-in-conversation-with-2023-1235523832/ &quot;Sylvester Stallone To Talk Career At Toronto 'In Conversation With…' Event&quot;]. ''[[Deadline Hollywood]]'', August 21, 2023.&lt;/ref&gt;
* 31 May 2023 &amp;ndash; [[Stallone]] move request to [[Stallone (surname)]] by [[User:BD2412|BD2412]] was ''moved'' to [[Sylvester Stallone]] by [[User:Skarmory|Skarmory]] on 07 Jun 2023; [[Talk:Stallone#Requested move 31 May 2023|discussion]]
* 31 May 2023 &amp;ndash; [[Stallone]] move request to [[Stallone (surname)]] by [[User:BD2412|BD2412]] was ''moved'' to [[Sylvester Stallone]] by [[User:Skarmory|Skarmory]] on 07 Jun 2023; [[Talk:Stallone#Requested move 31 May 2023|discussion]]
As a surgeon, Marmureanu has worked for Warren Beatty, Leonardo DiCaprio, Paris Hilton, Jack Nicholson, Jason Statham, Steven Spielberg, Silvester Stallone, and Bruce Willis.&lt;ref name=&quot;auto2&quot;/&gt;&lt;ref name=&quot;auto3&quot;/&gt;
    <redirect title="Sylvester Stallone" />
      <comment>Retargeting to Sylvester Stallone per RM discussion at [[Special:Permalink/1158933837#Requested move 31 May 2023|Talk:Stallone (surname)]].</comment>
      <text bytes="98" sha1="ige0r1aariw03nlc792p3myjd5k5jaz" xml:space="preserve">#REDIRECT [[Sylvester Stallone]]
* [[Sylvester Stallone]]
In May 2023, It was reported that [[Maria Bakalova]] would be starring in the film, to be produced by [[Sylvester Stallone]] and Balboa Productions.&lt;ref&gt;{{cite web|url=https://variety.com/2023/film/markets-festivals/borat-2-maria-bakalova-my-masterpiece-sylvester-stallone-balboa-productions-1235618793/amp/|date= May 19, 2023|title='Borat 2' Breakout Maria Bakalova Starring in 'My Masterpiece' for Sylvester Stallone's Balboa Productions (EXCLUSIVE)|first=Brent |last=Lang|first2=Matt|last2=Donnelly|website=Variety|accessdate=4 July 2023}} &lt;/ref&gt;
* [[Sylvester Stallone]]
'''''The Family Stallone''''' is an American [[Reality television|reality]] television series produced by [[MTV Entertainment Studios]] and [[Bunim/Murray Productions]]. It premiered on [[Paramount+]] on May 17, 2023.&lt;ref name=&quot;wrap&quot;&gt;{{cite news |last1=Cobb |first1=Kayla |title='The Family Stallone' Scores Season 2 Renewal |url=https://www.thewrap.com/the-family-stallone-paramount-plus-renewed-season-2/ |access-date=26 May 2023 |work=[[TheWrap]] |date=24 May 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite news |last1=Garner |first1=Glenn |title=Watch Sylvester Stallone Cuddle a Puppy and Catch Up with Al Pacino in The Family Stallone Trailer |url=https://people.com/tv/sylvester-stallone-reality-show-the-family-stallone-premiere-date/ |access-date=26 May 2023 |work=[[People (magazine)|People]] |date=April 11, 2023 |language=en}}&lt;/ref&gt; On May 24, 2023, it was renewed for a second season,&lt;ref name=&quot;wrap&quot; /&gt;&lt;ref&gt;{{cite news |last1=Porter |first1=Rick |title='The Family Stallone' Renewed at Paramount+ |url=https://www.hollywoodreporter.com/tv/tv-news/family-stallone-renewed-season-2-paramount-plus-1235499356/ |access-date=26 May 2023 |work=[[The Hollywood Reporter]] |date=24 May 2023}}&lt;/ref&gt; which premiered on February 21, 2024.&lt;ref&gt;{{cite web|title=''The Family Stallone'': Paramount+ Sets Return Date For Second Season Of Docuseries|url=https://deadline.com/2024/01/the-family-stallone-second-season-premiere-1235794926/|last=Rice|first=Lynette|website=Deadline Hollywood|date=January 17, 2024|access-date=January 17, 2024}}&lt;/ref&gt;
*[[Sylvester Stallone]]
*[[Jennifer Flavin]]&lt;ref&gt;{{cite news |last1=Longeretta |first1=Emily |title=Sylvester Stallone Lands Paramount+ Reality Show 'The Family Stallone' With Wife Jennifer Flavin and Three Daughters |url=https://variety.com/2023/tv/news/sylvester-stallone-family-reality-show-paramount-plus-1235510037/ |access-date=26 May 2023 |work=[[Variety (magazine)|Variety]] |date=2 February 2023}}&lt;/ref&gt;
[[Category:Sylvester Stallone]]
He also directed the [[Elvis Presley]] documentary film ''[[Elvis Presley: The Searcher]]'',&lt;ref&gt;{{cite news|last=Perry|first=Elle|title=Priscilla Presley, David Porter to premiere Elvis HBO film at SXSW|url=https://www.bizjournals.com/memphis/news/2018/03/01/priscilla-presley-david-porter-to-premiere-elvis.html|access-date=March 8, 2020|work=[[Memphis Business Journal]]|publisher=American City Business Journals|date=March 1, 2018}}&lt;/ref&gt; the [[Willie Nelson]] documentary series ''Willie Nelson and Family'',&lt;ref&gt;Hilton Dresden, [https://www.hollywoodreporter.com/movies/movie-news/willie-nelson-open-to-biopic-thom-zimny-oren-moverman-1235307526/ &quot;Willie Nelson Is "Very Open to a Biopic," Documentary Directors Say&quot;]. ''[[The Hollywood Reporter]]'', January 23, 2023.&lt;/ref&gt; and the 2023 [[Sylvester Stallone]] documentary film ''[[Sly (film)|Sly]]''.&lt;ref&gt;[https://www.theglobeandmail.com/arts/film/tiff/article-netflixs-sylvester-stallone-documentary-sly-to-close-tiff/ &quot;Netflix's Sylvester Stallone documentary Sly to close TIFF&quot;]. ''[[The Globe and Mail]]'', August 8, 2023.&lt;/ref&gt;
|title=Razzie Awards 2024: 'Winnie the Pooh' horror movie, Sylvester Stallone, Megan Fox take top 'prizes'
** [[Sylvester Stallone]] - ''[[Tulsa King]]'' as Dwight &quot;The General&quot; Manfredi ([[Paramount+]])
Singer has represented clients including [[Sylvester Stallone]], [[Scarlett Johansson]],&lt;ref name=&quot;:4&quot; /&gt; [[Arnold Schwarzenegger]],&lt;ref name=&quot;:5&quot; /&gt; [[Bill Cosby]],&lt;ref&gt;{{Cite web |last=Patten |first=Dominic |date=2015-10-20 |title=Bill Cosby Cuts Ties With Marty Singer &amp; Gets New Lawyer For Sex Assault Lawsuit |url=https://deadline.com/2015/10/bill-cosby-marty-singer-new-lawyer-judy-huth-sex-assault-lawsuit-1201589215/ |access-date=2023-08-07 |website=Deadline |language=en-US}}&lt;/ref&gt; [[Bruce Willis]],&lt;ref&gt;{{Cite web |last=Seitz |first=Matt Zoller |date=2022-08-03 |title=The Making of Silent Bruce |url=https://www.vulture.com/2022/08/silent-bruce-willis.html |access-date=2023-08-07 |website=Vulture |language=en-us}}&lt;/ref&gt; [[Charlie Sheen]], [[Harry Reid]],  [[Michael Jackson]], [[Oprah Winfrey]], [[Dennis Rodman]], [[Sheldon Adelson]], [[George Soros]], [[Tom Arnold (actor)|Tom Arnold]], [[Mike Myers]],&lt;ref name=&quot;:7&quot;&gt;{{Cite web |last=Margolick |first=David |date=2017 |title=&quot;Get Me Marty Singer!&quot; |url=https://archive.vanityfair.com/article/2017/3/get-me-marty-singer |access-date=2023-08-07 |website=Vanity Fair |language=en-US}}&lt;/ref&gt; [[Kate Gosselin]],&lt;ref name=&quot;:0&quot; /&gt; [[Britney Spears]],&lt;ref name=&quot;:0&quot; /&gt; [[Tiger Woods]],&lt;ref&gt;{{Cite web |last=Smith |first=Erin Geiger |title=Meet Tiger's Hush-Money Negotiators: Lavely &amp; Singer |url=https://www.businessinsider.com/who-are-tigers-lawyers-lavely-and-singer-not-new-to-spotlight-2009-12 |access-date=2023-08-07 |website=Business Insider |language=en-US}}&lt;/ref&gt;  and [[Jim Carrey]],&lt;ref name=&quot;:5&quot; /&gt; among others. He claims that his firm has filed more lawsuits against the media than any other law firm. His law firm Lavely &amp; Singer has been described as &quot;one of Hollywood's go-to law firms&quot;.&lt;ref name=&quot;:0&quot; /&gt;
Singer successfully represented photographer Max Aguilera-Hellweg in a lawsuit against Arnold Schwarzenegger, who was accused of using a photo without permission for a [[Bodybuilding|body-building]] calendar. His victory caught the attention of Schwarzenegger's lawyer, Jake Bloom, who then referred both Schwarzenegger and Sylvestor Stallone to Singer for assistance with their legal issues. He would go on to assist Schwarzenegger in two lawsuits brought by women claiming they were smeared by political aides, with one suit eventually settled and the other dismissed.&lt;ref name=&quot;:2&quot; /&gt;
Singer gained [[Tabloid journalism|tabloid]] attention in 1999 when [[Sylvester Stallone]] enlisted his services to counter a lawsuit filed by five former household employees. The workers claimed they experienced &quot;unusual directives&quot; from Stallone and [[Jennifer Flavin]], leading to their termination without full compensation, prompting a demand for $1.5 million in damages. One peculiar claim was an order to avoid communication or eye contact with Stallone's mother, Jackie. The staff, however, broke this rule when Jackie gifted them pens and autographed photos of herself for Christmas. Stallone and Flavin allegedly fired them upon discovering the breach. Singer vigorously refuted the allegations, stating they aimed to exploit the celebrity factor. He further contended that the workers were employed through a temporary agency on short-term contracts and highlighted the claim that one of the employees wore high heels while cleaning the house.&lt;ref name=&quot;:6&quot;&gt;{{Cite news |last=White |first=Adam |date=2017-11-08 |title=Hollywood's master of denial: meet Marty Singer, the lawyer who makes celebrity problems go away |language=en-GB |work=The Telegraph |url=https://www.telegraph.co.uk/films/0/hollywoods-master-denial-meet-marty-singer-lawyer-makes-celebrity/ |access-date=2023-08-07 |issn=0307-1235}}&lt;/ref&gt;
| budget         = $80 million&lt;ref&gt;{{Cite web|url=https://deadline.com/2017/05/jackie-chan-sylvester-stallone-ex-baghdad-scott-waugh-chinese-coproduction-1202090929/|title=Jackie Chan, Sylvester Stallone Team For Scott Waugh-Directed 'Ex-Baghdad'|first=Mike Jr.|last=Fleming|date=12 May 2017}}&lt;/ref&gt;
The movie was originally planned as a collaboration between Jackie Chan and [[Sylvester Stallone]], but Stallone backed out and was replaced by John Cena.&lt;ref name=empire&gt;{{cite web|url=https://www.empireonline.com/movies/news/john-cena-jackie-chan-team-up-action-comedy-trailer-hidden-strike/|title=John Cena And Jackie Chan Team Up In Trailer For Action-Comedy Hidden Strike|website=Empire}}&lt;/ref&gt;&lt;ref&gt;{{Cite web|url=https://variety.com/2018/film/news/john-cena-jackie-chan-project-x-1202843855/|title=John Cena to Replace Sylvester Stallone in Jackie Chan's Action-Thriller 'Project X'|first=Dave|last=McNary|date=12 June 2018}}&lt;/ref&gt; It was filmed in 2018,&lt;ref&gt;{{cite web|url=https://www.koimoi.com/hollywood-news/john-cena-once-shared-he-fell-into-depression-after-doing-a-film-with-jackie-chan-it-was-like-im-losing-everything-that-i-worked-on-for-30-years/|title=John Cena Once Shared He Fell Into Depression After Doing A Film With Jackie Chan: &quot;It Was Like I'm Losing Everything That I Worked On For 30 Years!&quot;|website=Koimoi}}&lt;/ref&gt; with locations in China standing in for the Middle East.&lt;ref name=collider1&gt;{{cite web|url=https://collider.com/john-cena-jackie-chan-movie-snafu-release-date-problems-explained/|title=John Cena and Jackie Chan Made a Movie Together That We May Never See|website=Collider|access-date=31 July 2023|archive-date=29 September 2023|archive-url=https://web.archive.org/web/20230929104345/https://collider.com/john-cena-jackie-chan-movie-snafu-release-date-problems-explained/|
```



```
url-status=dead}}&lt;/ref&gt;
|style=&quot;text-align:center;&quot; |&lt;ref&gt;{{Cite news|url=https://www.lemonde.fr/disparitions/article/2024/02/13/alain-dorval-voix-francaise-de-sylvester-stallone-et-de-nick-nolte-est-mort-a-77-ans_6216327_3382.html|title=Alain Dorval, voix française de Sylvester Stallone et de Nick Nolte, est mort à 77 ans|date=2024-02-13|access-date=2024-06-15|website=[[Le Monde]]}}&lt;/ref&gt;
|ShortSummary=A profile of [[Chuck Wepner]], the original inspiration for [[Sylvester Stallone]]'s [[Rocky Balboa]] character, and how ''[[Rocky]]''-like glory eluded Wepner as he took several strange turns in an effort to stay in the spotlight.
It was released on September 9, 2022, by [[Amazon Studios]].&lt;ref&gt;{{cite web|url=https://deadline.com/2022/08/amazon-movies-release-dates-harry-styles-my-policeman-sylveter-stallone-samaritan-1235082314/|title=Prime Video Dates Fall Movie Slate: Harry Styles' 'My Policeman', Lena Dunham's 'Catherine Called Birdy', Sylvester Stallone's 'Samaritan', 'Good Night Oppy' &amp; More|website=Deadline Hollywood|first=Anthony|last=D'Alessandro|date=August 1, 2022|access-date=August 20, 2023}}&lt;/ref&gt;
::: &quot;if a name cannot be verified in reliable sources ... then it has no place in that biography&quot; is true, but wouldn't seem to amount to a revdel/OS rationale, which is what this seem to be about. E.g., I could get high and go vandalize the [[Sylvester Stallone]] article to say that his birth name was &quot;Yojimbo Z. Doodah&quot;, but that would not be worth doing anything about other than a revert (and maybe a short block). It seems to be that there has to be both a credible claim that the name is legit, and a credible claim that there is an unusual privacy reason to suppress it. &lt;span style=&quot;white-space:nowrap;font-family:'Trebuchet MS'&quot;&gt;-[[User:SMcCandlish|'''SMcCandlish''']] [[User talk:SMcCandlish|☏]] [[Special:Contributions/SMcCandlish|¢]] 😼 &lt;/span&gt; 21:02, 26 September 2023 (UTC)
Fowler attended [[American University]] from 1976 to 1977, but left school become an announcer for the [[San Antonio Missions|San Antonio Dodgers]]. In 1980, he became the weekend sports anchor at San Antonio's KSAT-TV, and quickly became the station's sports director. In 1984 he moved to [[New York City]] to pursue an acting career.&lt;ref name=&quot;Chad&quot; /&gt; He appeared in the [[Sylvester Stallone]] film ''[[Cobra (1986 film)|Cobra]]'' and on [[HBO]]'s ''[[1st &amp; Ten (1984 TV series)|First and Ten]]''.&lt;ref name=&quot;Goff&quot; /&gt;
:Thanks for catching the hanging sentence Launchballer; fixed now. As for Doug Poppa, let's grant him that apparently he takes a strong interest in Las Vegas law enforcement, with a special focus on actual police reports and court documents. Black Kite and Schwede66, even if you might recoil a bit from citing him, I don't see why we should not be able to at least attribute his statements. After all, he thoroughly covered every single angle of this case over a period of seven years. He also uncovered a Sylvester Stallone Las Vegas [[Sylvester Stallone#Legal issues|hotel room liaison]] just in time for [[#MeToo]], and it was good enough for ''[https://www.usatoday.com/story/life/people/2017/11/16/rep-sexual-assault-claim-against-sylvester-stallone-categorically-false/872915001/ USA Today]'' to do a follow-up on. As I've said above, his personal musings and/or other interests have absolutely no bearing upon, nor have any place in, this article. &quot;[[Just the facts, Ma'am]]&quot;. '''[[User:StonyBrook|&lt;span style=&quot;color: #617aa3&quot;&gt;Stony&lt;/span&gt;&lt;span style=&quot;color: darkblue;&quot;&gt;Brook&lt;/span&gt;]]''' &lt;sub&gt;[[User talk:StonyBrook|''babble'']]&lt;/sub&gt; 07:22, 16 January 2024 (UTC)
In 1985, it was reported that Friedkin was to have directed [[Sylvester Stallone]] in a film adaptation of [[Don Pendleton]]'s book series ''[[The Executioner (book series)|The Executioner]]''.&lt;ref name=chapman/&gt;
      &quot;Sylvester Stallone&quot;: {
:You would have to first come up with a list of &quot;stars&quot; (however that's defined) and then look for their first known appearance in a movie. For one example, see [[Sylvester Stallone filmography]]. ←[[User:Baseball Bugs|Baseball Bugs]] &lt;sup&gt;''[[User talk:Baseball Bugs|What's up, Doc?]]''&lt;/sup&gt; [[Special:Contributions/Baseball_Bugs|carrots]]→ 22:45, 8 November 2023 (UTC)
| starring       = [[Sylvester Stallone]]
'''''Sly''''' is a 2023 American [[documentary film]] about the life and legacy of [[Sylvester Stallone]], directed by [[Thom Zimny]]. It premiered at the [[2023 Toronto International Film Festival]], and was released on [[Netflix]] on November 3, 2023.
The film explores Sylvester Stallone's life and nearly 50-year career, from rough childhood in Manhattan to struggling actor to action filmmaker and star of Hollywood franchises including ''[[Rocky (franchise)|Rocky]]'' and ''[[Rambo (franchise)|Rambo]]''.&lt;ref name=&quot;deadline&quot;/&gt;&lt;ref name=&quot;variety&quot;&gt;{{cite news |last1=Jackson |first1=Angelique |title=Sylvester Stallone Admits Career 'Regrets' in Trailer for Netflix Documentary 'Sly': 'Filmmaking, It Comes at a Great Price' |url=https://variety.com/2023/film/news/sylvester-stallone-netflix-documentary-sly-1235658671/ |access-date=25 October 2023 |publisher=Variety |date=30 June 2023}}&lt;/ref&gt; It includes footage of Stallone's pursuit of filmmaking, scenes from his films, photos from his childhood, and interviews with [[Arnold Schwarzenegger]], [[Frank Stallone]], [[Henry Winkler]], [[Talia Shire]], [[John Herzfeld]], [[Wesley Morris]], and [[Quentin Tarantino]].&lt;ref name=&quot;indiewire&quot;/&gt;&lt;ref&gt;{{cite news |last1=Pettibone |first1=Kat |title=Everything to Know About Sylvester Stallone's 'Sly' Documentary |url=https://www.usmagazine.com/entertainment/news/sylvester-stallones-sly-documentary-everything-to-know/ |access-date=25 October 2023 |publisher=Us |date=20 October 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite news |last1=Smart |first1=Jack |title=Sylvester Stallone Talks Fame and Life Regrets in Revealing Sly Documentary Trailer |url=https://people.com/see-sylvester-stallone-in-sly-documentary-trailer-exclusive-8362340 |access-date=25 October 2023 |publisher=People |date=18 October 2023}}&lt;/ref&gt;
The film premiered at the [[2023 Toronto International Film Festival]] as the closing night film on September 16, 2023,&lt;ref name=&quot;deadline&quot;&gt;{{cite news |last1=D'Alessandro |first1=Anthony |title=TIFF Closing Night Gala Is Netflix Sylvester Stallone Docu 'Sly' |url=https://deadline.com/2023/08/tiff-closing-night-sylvester-stallone-sly-documentary-1235457290/ |access-date=25 October 2023 |publisher=Deadline |date=8 August 2023}}&lt;/ref&gt; and was released on [[Netflix]] on November 3, 2023.&lt;ref name=&quot;indiewire&quot;&gt;{{cite news |last1=Bergeson |first1=Samantha |title=Sylvester Stallone Questions the Illusion of Success in Documentary About His Legacy |url=https://www.indiewire.com/news/trailers/sly-trailer-sylvester-stallone-questions-success-legacy-doc-1234918067/ |access-date=20 October 2023 |publisher=Indiewire |date=18 October 2023}}&lt;/ref&gt;
{{Rotten Tomatoes prose|81|6.4|54|A thoroughly watchable testament to Sylvester Stallone's longevity, ''Sly'' might pull its punches, but it's consistently engaging regardless.}}&lt;ref&gt;{{cite web|url=https://www.rottentomatoes.com/m/sly_2023|title=Sly|website=[[Rotten Tomatoes]]|access-date=16 February 2024}}&lt;/ref&gt; {{Metacritic film prose|62|17}}&lt;ref&gt;{{cite web|url=https://www.metacritic.com/movie/sly-2023/|title=Sly|website=[[Metacritic]]|access-date=16 November 2023}}&lt;/ref&gt;
{{Sylvester Stallone}}
[[Category:Sylvester Stallone]]
Taylor co-wrote ''[[Creed II]]'' (2018) alongside [[Sylvester Stallone]], featuring [[Michael B. Jordan]] and directed by [[Steven Caple Jr.|Steve Caple Jr]]. He also co-wrote ''[[Space Jam: A New Legacy]]'' (2021) with his writing partner [[Tony Rettenmaier]], starring [[LeBron James]] and directed by [[Malcolm D. Lee]].&lt;ref&gt;{{Citation |title=Space Jam: A New Legacy (2021) - IMDb |url=http://www.imdb.com/title/tt3554046/fullcredits |access-date=2024-03-05}}&lt;/ref&gt;
|[[File:Sylvester Stallone (210610105).jpg|100px]]
| Netflix released ''Sly'', a documentary about action movie legend [[Sylvester Stallone]] (pictured), and yet the man himself got less views than his first son, who also acted - including as [[Rocky Balboa]]'s son in ''[[Rocky V]]'' - and died at just 36 in 2012 of a heart disease.
| [[Sylvester Stallone]]
|[[File:Sylvester Stallone 2008.JPG|100px]]
Multiple celebrities such as [[John Wayne]], [[Reba McEntire]], [[Toby Keith]], and [[Ronald Reagan]] have visited the restaurant.&lt;ref name=&quot;Provine McCoy 2016 p. 66&quot; /&gt; Other celebrities like [[Sylvester Stallone]], [[Shaquille O'Neal]], and [[Charles Barkley]] also have dined at the restaurant.&lt;ref name=&quot;Griffin 2023 g707&quot; /&gt; [[George H. W. Bush]] also ate at the restaurant.&lt;ref name=&quot;ROUTE Magazine h848&quot;&gt;{{cite web |title=Historic Cattlemen's Steakhouse in Oklahoma |url=https://www.routemagazine.us/stories/cattlemens-cafe |access-date=2023-11-09 |website=ROUTE Magazine}}&lt;/ref&gt;
* 13 February : [[Alain Dorval]], 77, French voice actor, he dubbed [[Sylvester Stallone]], from 1976 to 2024
                                          &quot;Sylvester Stallone&quot;,
[[File:Sylvester Stallone Cannes 2019.jpg|thumb|right|150px|[[Sylvester Stallone]], Worst Supporting Actor winner]]
* '''[[Sylvester Stallone]] - ''[[Expend4bles]]'' as Barney Ross'''
* [[Sylvester Stallone]]
| starring       = Sylvester Stallone
'''''Cliffhanger 2''''' is an upcoming American [[action thriller film]], directed by [[Jean-François Richet]] and written by Mark Bianculli. It serves as the sequel to the 1993 film ''[[Cliffhanger (film)|Cliffhanger]]''. [[Sylvester Stallone]] reprises his role as Gabriel &quot;Gabe&quot; Walker.
* [[Sylvester Stallone]] as Ranger Gabriel &quot;Gabe&quot; Walker, a former mountain climber and rescue ranger
In May 2023, it was announced that a sequel to the 1993 film ''[[Cliffhanger (film)|Cliffhanger]]'' was in development, with [[Ric Roman Waugh]] directing the film, and [[Sylvester Stallone]] returning.&lt;ref name=&quot;CH2&quot; /&gt; In December, it was revealed that Waugh had exited the project, and that [[Jean-François Richet]] had been hired as the director of the film, with Mark Bianculli writing the screenplay. [[Principal photography]] is scheduled to begin in fall 2024 in [[Bavaria]] and [[Munich]].&lt;ref name=&quot;Richet&quot; /&gt;
&lt;ref name=&quot;CH2&quot;&gt;{{Cite web |last=Perlata |first=Diego |date=May 1, 2023 |title=Sylvester Stallone Is Returning for a 'Cliffhanger' Reboot |url=https://collider.com/sylvester-stallone-cliffhanger-reboot/ |url-status=live |archive-url=https://web.archive.org/web/20230630161929/https://collider.com/sylvester-stallone-cliffhanger-reboot/ |archive-date=June 30, 2023 |access-date=January 4, 2024 |website=[[Collider]]}}&lt;/ref&gt;
{{Sylvester Stallone}}
*[[Alain Dorval]], 77, French voice actor ([[Sylvester Stallone]], [[Nick Nolte]], [[Pete (Disney)|Pete]]), cancer.&lt;ref&gt;[https://www.leparisien.fr/culture-loisirs/cinema/alain-dorval-voix-francaise-de-sylvester-stallone-et-pere-de-la-ministre-aurore-berge-est-mort-a-77-ans-13-02-2024-LBRHLKLMV5EZXE4PCFFCEHLJYY.php Alain Dorval, voix française de Sylvester Stallone et père de la ministre Aurore Bergé, est mort à 77 ans] {{in lang|fr}}&lt;/ref&gt;
* [[Sylvester Stallone]]
'''''Armored''''' is an upcoming American [[action thriller film]] directed by Justin Routt and starring [[Sylvester Stallone]].
* [[Sylvester Stallone]]&lt;ref&gt;{{cite web|title=Stallone movie looking for stand-in|date=September 26, 2023|website=WXXV-TV|url=https://www.wxxv25.com/stallone-movie-looking-for-stand-in/|accessdate=January 2, 2024}}&lt;/ref&gt;
* [[Jason Patric]]&lt;ref&gt;{{cite web|last=Parker|first=Brooke|title=Movie starring Sylvester Stallone and Jason Patric being filmed in Waveland|date=September 21, 2023|website=WXXV-TV|url=https://www.wxxv25.com/movie-starring-sylvester-stallone-and-jason-patrick-being-filmed-in-waveland/|accessdate=January 2, 2024}}&lt;/ref&gt;
* [[Josh Wiggins]]&lt;ref&gt;{{cite news|last=Mitchell|first=Justin|title=Could you be Sylvester Stallone's twin brother? A movie filming in South MS needs you.|date=September 26, 2023|newspaper=Sun
```



```
Herald|url=https://www.sunherald.com/entertainment/article279774064.html|accessdate=January 2, 2024}}&lt;/ref&gt;
* [[Dash Mihok]]&lt;ref&gt;{{cite news|last=Sanchez|first=Martha|title=Action movie starring Sylvester Stallone is filming in two MS Coast cities this month|date=September 21, 2023|newspaper=Sun Herald|url=https://amp.sunherald.com/entertainment/article279603509.html|accessdate=January 2, 2024}}&lt;/ref&gt;
Filming occurred in [[Pearlington, Mississippi]] and [[Waveland, Mississippi]] in September 2023.&lt;ref&gt;{{cite web|last=Kulo|first=Warren|title=Sylvester Stallone filming new movie on Mississippi coast|date=September 22, 2023|website=Al.com|url=https://www.al.com/news/2023/09/sylvester-stallone-filming-new-movie-on-mississippi-coast.html|accessdate=December 30, 2023}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last=Sanchez|first=Martha|title=Action movie starring Sylvester Stallone is filming in two MS Coast cities this month|date=September 21, 2023|website=Yahoo! News|url=https://news.yahoo.com/action-movie-starring-sylvester-stallone-174322185.html|accessdate=December 30, 2023}}&lt;/ref&gt;  Since filming occurred during the [[2023 SAG-AFTRA strike]], the filmmakers were granted interim agreements which gave them permission to make the film.&lt;ref&gt;{{cite web|last=Spradley|first=Amber|title=Crew shoots Sylvester Stallone movie on coast amid strike adhering to interim agreements|date=September 21, 2023|website=WLOX|url=https://www.wlox.com/2023/09/22/crew-shoots-sylvester-stallone-movie-coast-amid-strike-adhering-interim-agreements/|accessdate=January 2, 2024}}&lt;/ref&gt;
[[Category:Images of Sylvester Stallone]]</text>
|Guests=[[Sylvester Stallone]], [[Jalen Brunson]]
::::...Fans of Sylvester Stallone/Rocky, I know. [[User:Gråbergs Gråa Sång|Gråbergs Gråa Sång]] ([[User talk:Gråbergs Gråa Sång|talk]]) 21:20, 16 December 2023 (UTC)
| screenplay     = [[Sylvester Stallone]]
* Sylvester Stallone
'''''Levon's Trade''''' is an upcoming American [[action thriller film]] directed by [[David Ayer]] and written by [[Sylvester Stallone]], based on the 2014 novel of the same name by [[Chuck Dixon]]. It stars [[Jason Statham]], [[David Harbour]], [[Michael Peña]] and [[Jason Flemyng]].
[[Sylvester Stallone]] originally developed an adaptation of the [[Chuck Dixon]] novel ''Levon's Trade'' as a television series with [[Balboa Productions]].&lt;ref&gt;{{cite web |last1=N'Duka |first1=Amanda |title=Sylvester Stallone To Star In ''Samaritan''&amp; ''Hunter'', Acquires FBI Memoir as his Balboa Productions Ramps Up Slate |url=https://deadline.com/2018/10/sylvester-stallone-samaritan-hunter-balboa-productions-ghost-book-deal-1202483446/ |website=[[Deadline Hollywood]] |publisher=[[Penske Media Corporation]] |date=October 17, 2018}}&lt;/ref&gt;  The project changed into a film that was up for sale at the 2023 [[American Film Market]], where the film was first announced to have [[David Ayer]] attached to direct and [[Jason Statham]] to star.&lt;ref name=&quot;AFM&quot;&gt;{{cite web |last1=Wiseman |first1=Andreas |title=Jason Statham To Star In David Ayer-Directed Sylvester Stallone Script 'Levon's Trade' From Black Bear &amp; Bill Block's BlockFilm – AFM Hot Package |url=https://deadline.com/2023/10/jason-statham-david-ayer-sylvester-stallone-levons-trade-black-bear-bill-block-afm-1235585741/ |website=[[Deadline Hollywood]] |publisher=[[Penske Media Corporation]] |date=October 27, 2023}}&lt;/ref&gt; In January 2024, [[Metro-Goldwyn-Mayer]] acquired U.S. distribution rights to the film from [[Black Bear Pictures|Black Bear International]], who sold the film to independent distributors elsewhere.&lt;ref&gt;{{cite web |last1=Wiseman |first1=Andreas |title=Amazon MGM Lands Jason Statham Action-Thriller 'Levon's Trade' With David Ayer Directing Sylvester Stallone Script; Wide Theatrical Release Lined Up In U.S. |url=https://deadline.com/2024/01/jason-statham-david-ayer-sylvester-stallone-amazon-levons-trade-movie-deal-1235805593/ |website=[[Deadline Hollywood]] |publisher=[[Penske Media Corporation]] |date=January 26, 2024}}&lt;/ref&gt; The film was put into development in part due to its potential for subsequent entries given the number of novels written by Dixon in the series.&lt;ref&gt;{{cite web |last1=O'Rourke |first1=Ryan |title=David Ayer Teases His Jason Statham-Led 'Levon's Trade' Adaptation |url=https://collider.com/jason-statham-levons-trade-david-ayer/ |website=[[Collider (website)|Collider]] |publisher=Valnet Inc. |language=en |date=January 5, 2024}}&lt;/ref&gt;
In April 2024, [[David Harbour]], [[Michael Peña]], [[Jason Flemyng]], Arianna Rivas, [[Noemi Gonzalez]], [[Emmett J. Scanlan]], [[Eve Mauro]], [[Maximilian Osinski]], Kristina Poli, Andrej Kaminsky and Isla Gie joined the cast in undisclosed roles.&lt;ref name=&quot;Apr'24 casting&quot;&gt;{{cite web |last1=Wiseman |first1=Andreas |title=Michael Peña, David Harbour, Jason Flemyng &amp; Arianna Rivas Join Jason Statham In David Ayer's 'Levon's Trade'; Amazon MGM Sets Release Date For Black Bear Action Pic Scripted By Sylvester Stallone |url=https://deadline.com/2024/04/david-ayer-jason-statham-slyvester-stallone-movie-levons-trade-begins-amazon-mgm-release-date-1235878520/ |website=[[Deadline Hollywood]] |publisher=[[Penske Media Corporation]] |date=April 8, 2024}}&lt;/ref&gt;
{{Sylvester Stallone}}
[[Category:Films with screenplays by Sylvester Stallone]]
In 2021 Irsay purchased [[Sylvester Stallone]]'s early working script notebook for the film ''[[Rocky]]'' (1976), written by and starring Stallone.&lt;ref name=&quot;Sly&quot;&gt;{{cite news |title=Sylvester Stallone 'Rocky' script notes find new home in Indiana |url=https://www.wthr.com/video/news/sylvester-stallone-rocky-script-notes-find-new-home-in-indiana/531-12d97a6f-a871-4e83-aed7-d3fc1784250c |access-date=20 December 2023 |work=WTHR |date=9 December 2021}}&lt;/ref&gt;
* [[Sylvester Stallone]] (2016)
In 2014, Gisella Marengo embarked on her journey as a film producer. She produced several projects, including &quot;[[The Humbling (film)|The Humbling]]&quot; starring [[Al Pacino]] and directed by [[Barry Levinson]],&lt;ref name=&quot;:0&quot; /&gt;and the American drama &quot;[[Blackbird (2019 film)|Blackbird]],&quot; featuring [[Susan Sarandon]] and [[Kate Winslet]], directed by [[Roger Michell]].&lt;ref name=&quot;:0&quot; /&gt; She also worked on action movies such as &quot;[[Angel Has Fallen]]&quot; with [[Gerard Butler|Gerald Butler]] and [[Morgan Freeman]],&lt;ref name=&quot;:0&quot; /&gt; &quot;[[Hunter Killer (film)|Hunter Killer]]&quot; starring [[Gerard Butler|Gerald Butler]] and [[Gary Oldman]],&lt;ref name=&quot;:0&quot; /&gt; and &quot;Security&quot; featuring [[Antonio Banderas]].&lt;ref name=&quot;:0&quot; /&gt;Among her recent productions, in 2020, Gisella Marengo produced  the documentary &quot;[[Francesco (2020 film)|Francesco]],&quot; directed by Oscar-nominated [[Evgeny Afineevsky]].&lt;ref name=&quot;:0&quot; /&gt; The film premiered at the [[Rome Film Festival]]&lt;ref name=&quot;:13&quot;&gt;{{Cite web |date=2021-03-26 |title=The film &quot;Francesco&quot; on Discovery+ - Vatican News |url=https://www.vaticannews.va/en/vatican-city/news/2021-03/documentary-film-francesco-afineevsky-pope-francis-streaming.html |access-date=2024-02-08 |website=www.vaticannews.va |language=en}}&lt;/ref&gt; and was awarded with the 18th Kinéo Prize.&lt;ref name=&quot;:14&quot;&gt;{{Cite web |date=2020-10-17 |title=Festa Roma: premio Kineo a Afineevsky per docufilm sul Papa - Cinema - Ansa.it |url=https://www.ansa.it/sito/notizie/cultura/cinema/2020/10/17/festa-roma-premio-kineo-a-afineevsky-per-docufilm-sul-papa_ab6a123e-e79b-4b03-8f9e-1cb3e1bcb432.html |access-date=2024-02-08 |website=Agenzia ANSA |language=it}}&lt;/ref&gt; The following year, she was the producer behind the highly acclaimed and renowned documentary &quot;[[Ennio (film)|Ennio]],&quot; which delved into the life of the famous Italian Oscar-winning film composer, [[Ennio Morricone]].&lt;ref name=&quot;:15&quot;&gt;{{Cite web |title=Prime Video: ENNIO THE MAESTRO |url=https://www.primevideo.com/detail/ENNIO-THE-MAESTRO/0O398U97308YFWR5YB0LSSLTUZ |access-date=2024-02-08 |website=www.primevideo.com |language=en-us}}&lt;/ref&gt; Directed by Oscar winning director [[Giuseppe Tornatore]], the documentary features interviews with  world-class artists, including [[Clint Eastwood]], [[Quentin Tarantino]], [[Hans Zimmer]], and [[Bruce Springsteen]].&lt;ref name=&quot;:15&quot; /&gt; It made its debut at the [[Venice Film Festival|Venice International Film Festival]]&lt;ref name=&quot;:16&quot;&gt;{{Cite web |date=2021-08-10 |title=Biennale Cinema 2021 {{!}} Ennio by Giuseppe Tornatore screens Out of Competition at the 78th Festival |url=https://www.labiennale.org/en/news/ennio-giuseppe-tornatore-screens-out-competition-78th-festival |access-date=2024-02-08 |website=La Biennale di Venezia |language=en}}&lt;/ref&gt; and it was awarded with the prestigious [[David di Donatello|David di Donatello Award]] for Best Documentary and the Starlight International Film Award.&lt;ref name=&quot;:17&quot;&gt;{{Cite web |date=2022-05-04 |title=Ai David di Donatello vola 'E' stata la mano di Dio' di Paolo Sorrentino - Notizie - Ansa.it |url=https://www.ansa.it/sito/notizie/cultura/2022/05/04/ai-david-di-donatello-vola-e-stata-la-mano-di-dio-di-paolo-sorrentino_63e3d407-8794-42a3-9480-3cd4a683d057.html |access-date=2024-02-08 |website=Agenzia ANSA |language=it}}&lt;/ref&gt; In 2023, Marengo co-produced the action-packed &quot;[[Expend4bles]],&quot; featuring actors like [[Megan Fox]], [[Sylvester Stallone]], and [[Jason Statham]].&lt;ref name=&quot;:0&quot; /&gt;
| 2022 || [[Tulsa King]] &lt;ref&gt;{{cite web |last1=Soto |first1=Josiah |title='Tulsa King' Is Newest Show from the Creator of 'Yellowstone' Starring Sylvester Stallone |url=https://www.thepioneerwoman.com/news-entertainment/a41721465/tulsa-king-season-1-date-cast/ |website=The Pioneer Woman |date=20 October 2022}}&lt;/ref&gt;|| Carson Pike ||
Duffy's designs are often featured in design publications and pieces have been exhibited at events including Art Basel, Art Miami, 100% Design, PAD London, and Clerkenwell Design Week.&lt;ref&gt;{{Cite web|url=https://www.clerkenwelldesignweek.com/pause-duffy-london|title=Pause - Duffy London|website=Clerkenwell Design Week 2024}}{{dead link|date=October 2024}}&lt;/ref&gt; They have been acquired by Sylvester Stallone, Prince, and Miley Cyrus, along with companies including Google, Hudson Yards, and Four Seasons Hotels.{{Citation needed|date=October 2024}}
Celebrities who has joined Joseph's programs&lt;ref&gt;{{Cite web |title=Voice |url=https://www.pressadvantage.com/story/8929-voice-coach-to-the-stars-arthur-samuel-joseph-releases-hollywood-vocal-secrets-to-the-public%%22 |access-date=2024-03-13 |website=Vocal}}&lt;/ref&gt; includes [[Arnold Schwarzenegger]], [[Charlton Heston]], [[Sean Connery]], [[Sylvester Stallone]], [[Angelina Jolie]], [[Sally Field]], [[Steven Seagal]], [[Aaron Boone]], [[Barry Larkin|Barry Larking]], [[Magic Johnson]], [[Dwyane Wade|Dwayne Wade]].
* [[Sylvester Stallone]]
'''''Alarum''''' is an upcoming American [[action film|action]] [[crime thriller film]] directed by [[Polish brothers|Michael Polish]] and starring [[Sylvester Stallone]], [[Scott Eastwood]], [[Willa Fitzgerald]], [[Mike Colter]], [[Ísis Valverde]], and [[D. W. Moffett]].&lt;ref&gt;{{cite web|last=Schramm|first=Molly|title=Sylvester Stallone filming two movies in the Greater Cincinnati area|date=March 2, 2024|website=WCPO-TV|url=https://www.wcpo.com/entertainment/local-a-e/local-movies/sylvester-stallone-filming-two-movies-in-the-greater-cincinnati-area|accessdate=March 18, 2024}}&lt;/ref&gt;
* [[Sylvester Stallone]] as Chester
The film was awarded $5,863,392 from the Ohio Motion Picture Tax Credit Program on February 13, 2024.&lt;ref&gt;{{cite web|last=Kiesewetter|first=John|title=Sylvester Stallone movie arriving in Oxford with a bang|date=February 22, 2024|website=WVXU|url=https://www.wvxu.org/media/2024-02-22/sylvester-stallone-alarum-movie-oxford-ohio-tvkiese|accessdate=March 18, 2024}}&lt;/ref&gt;&lt;ref&gt;{{cite web|last=Naquin|first=Talia|title=Details on Sylvester Stallone movie filming in Ohio|date=February 23, 2024|website=Yahoo! News|url=https://news.yahoo.com/details-sylvester-stallone-movie-filming-182328783.html?guccounter=1|accessdate=March 18, 2024}}&lt;/ref&gt;
[[Principal photography]] began on February 26 in [[Oxford, Ohio]],&lt;ref&gt;{{cite web|title=Filming underway for Sylvester Stallone movie in Ohio|date=February 26, 2024|website=WHIO-TV|url=https://www.whio.com/news/state-and-regional/filming-underway-sylvester-stallone-movie-ohio/5DUHAF7WBJF2NDGQCSMVCTXADI/|accessdate=March 18, 2024}}&lt;/ref&gt; and [[wrap (filmmaking)|wrapped]] on May 14.&lt;ref&gt;{{cite web|first=Zac |last=Ntim |title='Alarum:' Shooting Wraps On Action Pic Starring Sylvester Stallone, Scott Eastwood &amp; Willa Fitzgerald, Highland Film Group Launches Sales
In May 2024, it was announced that [[Signature Entertainment]] acquired U.K. and Irish distribution rights to the film.&lt;ref&gt;{{cite magazine|last=Ritman|first=Alex|title=Sylvester Stallone Action Thriller 'Alarum' Acquired by Signature for U.K. (EXCLUSIVE)|date=May 17, 2024|magazine=Variety|url=https://variety.com/2024/film/global/sylvester-stallones-alarum-signature-uk-1236005945/amp/|accessdate=May 17, 2024}}&lt;/ref&gt;
On television, he featured in [[Amazon Prime Video]] series ''[[The Marvelous Mrs. Maisel]]'' and had a main role in the first series of [[Taylor Sheridan]] and [[Sylvester Stallone]] series ''[[Tulsa King]]'' on ''[[Paramount+]]'' as Tyson.&lt;ref&gt;{{cite web|url=https://deadline.com/2022/03/tulsa-king-max-casella-domenick-lombardozzi-vincent-piazza-jay-cast-sylvester-stallone-paramount-plus-1234985490/|title='Tulsa King': Max Casella, Domenick Lombardozzi, Vincent Piazza &amp; Jay Will Join Sylvester Stallone In Paramount+ Series|last=Andreeva|first=Nellie|date=March 24, 2022|accessdate=May 16, 2024|website=Deadline Hollywood}}&lt;/ref&gt;&lt;ref&gt;{{cite web|url= https://collider.com/tulsa-king-tv-show-cast-max-casella-jay-will-sylverster-stallone/|website=Collider|accessdate=17 May 2024|title= 'Tulsa King' Adds Max Casella, Jay Will, and More to Sylvester Stallone Mob Series|first=Lacy|last=Long|date=24 March 2022}}&lt;/ref&gt;
```



3539 {{block indent|text=Coca Cola Collection, Verona, Italy; Deutsche Bank Stiftung, Frankfurt, Germany; Groninger Museum, Groningen, Netherlands; Hessisches Landesmuseum Darmstadt, Germany; Museum Würth, Künzelsau, Germany; Kunstmuseum Düsseldorf, Germany; Kunstmuseum Bonn, Germany; Louis Vuitton, Italy / France; Sylvester Stallone, Santa Monica, USA; Stadtmuseum Düsseldorf, Germany; Kunsthalle Mannheim, Germany; Rooseum Malmö, Sweden; MOCA Museum of Contemporary Art, Los Angeles, USA; Museum Boijmans Van Beuningen, Rotterdam, Netherlands; Stedelijk Museum, Amsterdam, Netherlands; Olomouc Museum of Art, Olomouc, Czech Republic; National Gallery Prague, Czech Republic; Fonds Hélène et Édouard Leclerc, Landerneau, France.}}

3540 '''Ralph DeLuca''' (born May 13, 1976) is an American art advisor and collector specializing in [[contemporary art]] and [[film memorabilia]] based in [[New York City]] and [[Las Vegas]]. His clients include individuals, corporations, foundations, and institutions. Amongst his clients are actors [[Leonardo DiCaprio]],&lt;ref&gt;{{Cite news |last1=Bahr |first1=Sarah |last2=Syckle |first2=Katie Van |last3=Westervelt |first3=Nina |last4=Pinyodoonyachet |first4=Jutharat |date=2024-04-06 |title=Leonardo DiCaprio, Tracee Ellis Ross and Gayle King Hit New York Art Parties |url=https://www.nytimes.com/2024/04/06/style/nyc-new-museum-art-gala.html |access-date=2024-04-21 |work=The New York Times |language=en-US |issn=0362-4331}}&lt;/ref&gt; [[Sylvester Stallone]],&lt;ref&gt;{{Cite web |date=2023-05-26 |title=Did Leo DiCaprio's Former Art Advisor Exploit the Star's Name? |url=https://observer.com/2023/05/leo-dicaprios-art-advisor-sets-the-record-straight/ |access-date=2024-04-21 |website=Observer |language=en-US}}&lt;/ref&gt; and co-founder of [[Tinder (app)|Tinder]], [[Sean Rad]].&lt;ref&gt;{{Cite web |title=Six Art World Power Couples You Should Know |url=https://www.culturedmag.com/article/2021/02/14/six-art-world-power-couples-you-should-know |access-date=2024-04-21 |website=www.culturedmag.com |language=en}}&lt;/ref&gt;

3541 Mercanti has organized events featuring notable figures such as [[Al Pacino]],&lt;ref&gt;{{cite news|url=https://www.thespec.com/news/al-pacino-to-headline-charity-fundraiser-in-stoney-creek/article_3e282b47-df4a-53b9-8e33-8aa81831fc80.html|title=Al Pacino to headline charity fundraiser in Stoney Creek|newspaper=The Hamilton Spectator |publisher=thespec.com|date=10 June 2010|accessdate=1 May 2024}}&lt;/ref&gt; [[Sylvester Stallone]],&lt;ref&gt;{{cite news|url=https://www.thespec.com/news/hamilton-region/sylvester-stallone-brings-rocky-and-rambo-to-hamilton/article_6c3b294b-9fba-5f64-9990-d4bdd38477ee.html|title=Sylvester Stallone brings Rocky and Rambo to Hamilton|newspaper=The Hamilton Spectator |publisher=thespec.com|date=23 February 2012|accessdate=1 May 2024}}&lt;/ref&gt; [[Wayne Gretzky]],&lt;ref&gt;{{cite web|url=https://toronto.citynews.ca/2017/10/17/wayne-gretzky-leaves-fans-disappointed-hamilton-fundraiser/|title=Wayne Gretzky leaves fans disappointed at Hamilton fundraiser|publisher=toronto.citynews.ca|date=17 October 2017|accessdate=1 May 2024}}&lt;/ref&gt; and others for different charitable causes.

3542 [[Sylvester Stallone]] and Braden Aftergood are among the executive producers through [[Balboa Productions]]. It is produced by Mark Lane of Tea Shop Productions and Kevin Sampson of White Star Productions. AGC Studios and BondIt Media Capital are financing the film. AGC chairman and CEO [[Stuart Ford (entertainment executive)|Stuart Ford]] will also produce. &lt;ref&gt;{{cite web|url= https://variety.com/2024/film/global/pierce-brosnan-amir-el-masry-agc-prince-naseem-hamed-giant-sylvester-stallone-1235971227/amp/|website=Variety|accessdate=15 April 2024|date= April 15, 2024| title=Pierce Brosnan, Amir El-Masry Join AGC's Biopic of Boxer Prince Naseem Hamed 'Giant,' Sylvester Stallone to Executive Produce|first=Leo|last=Barraclough}}&lt;/ref&gt;

3543 &lt;li&gt;{{cite news |last=Pasta |first=Elmer |date=1996-12-13 |title=Sylvester Stallone resurrects his hero role in disastrous 'Daylight' |url=https://www.newspapers.com/article/hawaii-tribune-herald/148804716/ |newspaper=[[Hawaii Tribune-Herald]] |via=[[Newspapers.com]] |accessdate=2024-06-06 |archiveurl=https://web.archive.org/web/20240606092349/https://www.newspapers.com/article/hawaii-tribune-herald/148804716/ |archivedate=2024-06-06 }}&lt;p&gt;The review notes: &quot;&quot;Space Marines&quot; (Republic, rated R for excessive violence, sex and swearing) has impressive special effects for a relatively low-budget, direct-to-video sci-fi epic. But, this poor man's &quot;Star Trek&quot; has little more to offer than a basic, good-vs.-bad-guys plot with lots of ham overacting. ... The film-titled Space Marines begin a rescue-and-arrest mission that quickly turns into a volatile political situation. Ordered to return to base, but aware of the mounting threat to the galaxy's survival, the elite Marine Corps stages its own vigilante plan. It's easy to guess the predictable outcome, which includes a daring mutiny and eventual pirate confrontation. There's much futuristic space hardware in the meandering movie, including a spacecraft resembling a giant electric shaver. Even with that, over all, &quot;Space Marines&quot; hasn't got much buzz.&quot;&lt;/li&gt;

3544 Nöger has participated in numerous solo and group exhibitions in Europe, Asia, North and South America. His works can be found in international museums and in private collections, for example in the art collections of [[Monika Grütters]], [[Nils Seethaler]], [[Reinhold Würth]], [[Halle Berry]], [[Sylvester Stallone]], [[Elton John]] and [[Bill Gates]].&lt;ref&gt;{{cite web |accessdate=2021-04-20 |lang=en |title=Udo Nöger |url=https://www.kostuikgallery.com/udo-noger/ |work=Kostuik Gallery}}&lt;/ref&gt;

3545 ''Tulsa King'' is a streaming television series that is notable for being Sylvester Stallone's scripted television debut. The series is pretty well known for Stallone's performance. It has also set a few viewership records and has been the subject of a little controversy during second season production. In 2023 the series was nominated for a Creative Arts Emmy. Throughout the last few months I have put quite a bit of work into this article expanding it from a lower end C class all the way to GA. I believe it's now comprehensive enough to meet FA requirements. This will be my third FAC, and hopefully my second successful FA. I look forward to addressing any comments or concerns. [[User:TheDoctorWho|&lt;span style=&quot;color:#0000ff;&quot;&gt;'''The'''&lt;/span&gt;&lt;span style=&quot;color:#0000ff&quot;&gt;'''Doctor'''&lt;/span&gt;&lt;span style=&quot;color:#0000ff;&quot;&gt;'''Who'''&lt;/span&gt;]] [[User talk:TheDoctorWho|(talk)]] 06:51, 4 May 2024 (UTC)

3546 |Italian-American hairdresser and father of [[Sylvester Stallone]]

3547 '''''The Expendables: Original Motion Picture Soundtrack''''' is the [[soundtrack]] to the 2010 film ''[[The Expendables (2010 film)|The Expendables]]'' directed by [[Sylvester Stallone]]. Featuring music composed by [[Brian Tyler]], the soundtrack consisted of 20 tracks which was released by Lionsgate Records on August 10, 2010.

3548 [[Godsmack]] vocalist [[Sully Erna]] was approached by Stallone himself to write a song for the film. Erna showed him a potential unfinished piece of &quot;[[Sinner's Prayer (song)|Sinners Prayer]]&quot; originally recorded for his debut album ''[[Avalon (Sully Erna album)|Avalon]]'' (2010);&lt;ref&gt;{{Cite web |last=DemonDust |date=2009-09-27 |title=Godsmack singer hooks up with Sylvester Stallone for movie soundtrack |url=https://www.metalrage.com/news/52034/godsmack-singer-hooks-up-with-sylvester-stallone-for-movie-soundtrack.html |url-status=dead |archive-url=https://web.archive.org/web/20091126074503/http://www.metalrage.com/news/52034/godsmack_singer_hooks_up_with_sylvester_stallone_for_movie_soundtrack.html/ |archive-date=2009-11-26 |access-date=2024-06-20 |website=Metalrage}}&lt;/ref&gt; Stallone liked it and wanted to use it in the film. However, during the film's post-production, the scene that &quot;Sinner's Prayer&quot; was originally meant to be used in was reworked and the song was taken off the film.&lt;ref&gt;{{Cite web |date=2010-04-12 |title=Godsmack Frontman: 'Our Sound Is Much More Mature Now' |url=http://motleycrue.rockzillatv.com/godsmack-frontman-our-sound-is-much-more-mature-now-apr-12-2010-blabbermouth-net/ |archive-url=https://web.archive.org/web/20110715195819/http://motleycrue.rockzillatv.com/godsmack-frontman-our-sound-is-much-more-mature-now-apr-12-2010-blabbermouth-net/ |archive-date=2011-07-15 |access-date=2024-06-20 |website=Rockzilla.tv |publisher=BlabberMouth}}&lt;/ref&gt; The American hard rock band [[Shinedown]] had recorded the song &quot;[[Diamond Eyes (Boom-Lay Boom-Lay Boom)]]&quot;, specifically for the film, but was not included. The song was used in the theatrical trailer and the track in its entirety was released on June 15, 2010.&lt;ref&gt;{{cite web |author= |date= |title=Diamond Eyes **Boom Lay Boom** SHINEDOWN -LIVE- 7.7.10 |url=https://www.youtube.com/watch?v=8HJJfFNtfLs |url-status=live |archive-url=https://ghostarchive.org/varchive/youtube/20211221/8HJJfFNtfLs |archive-date=2021-12-21 |accessdate=February 13, 2012 |publisher=YouTube}}{{cbignore}}&lt;/ref&gt;&lt;ref&gt;{{cite web |title=Diamond Eyes (Boom-Lay Boom-Lay Boom) |url=http://www.atlanticrecords.com/news/article/?articleId=8a0af812274dc9b00127b9dcbc3a48b0 |url-status=live |archive-url=https://web.archive.org/web/20110925035415/http://www.atlanticrecords.com/news/article/?articleId=8a0af812274dc9b00127b9dcbc3a48b0 |archive-date=2011-09-25 |accessdate=February 13, 2012 |publisher=Atlantic Records}}&lt;/ref&gt; Both songs were finally used for the Extended Director's Cut.&lt;ref&gt;{{cite web |date=2010-04-01 |title=Shinedown's Unreleased Song Featured In The Expendables Movie Trailer! |url=http://www.atlanticrecords.com/news/article/?articleId=8a0af812274dc9b00127b9dcbc3a48b0 |url-status=live |archive-url=https://web.archive.org/web/20110815041934/http://www.atlanticrecords.com/news/article/?articleId=8a0af812274dc9b00127b9dcbc3a48b0 |archive-date=2011-08-15 |access-date=2010-04-02 |publisher=Atlantic Records}}&lt;/ref&gt; One of the alternate trailers uses the song &quot;[[Paradise City]]&quot; by [[Guns N' Roses]]. The song &quot;[[The Boys Are Back in Town]]&quot; by [[Thin Lizzy]] (live version featured on ''[[Still Dangerous]]'') played in TV spots and is played over the credits.&lt;ref&gt;{{Cite news |last=Heritage |first=Stuart |date=2010-07-07 |title=The Expendables: acting their age |url=https://www.theguardian.com/film/filmblog/2010/jul/07/expendables-arnold-schwarzenegger-sylvester-stallone |url-status=live |archive-url=https://web.archive.org/web/20130915145404/https://www.theguardian.com/film/filmblog/2010/jul/07/expendables-arnold-schwarzenegger-sylvester-stallone |archive-date=2013-09-15 |access-date=2024-06-20 |work=The Guardian |language=en-GB |issn=0261-3077}}&lt;/ref&gt;

3549 [[File:Sylvester Stallone Cannes 2019.jpg|thumb|120px|Sylvester Stallone]]

3550 | [[Sylvester Stallone]] || Michael Sylvester Stallone || style=&quot;text-align: center;&quot; |&lt;ref name=&quot;slystallone&quot;/&gt;&lt;ref name=&quot;mitchell 2007&quot;/&gt;&lt;ref name=&quot;middleroad&quot;/&gt;

3551 &lt;ref name=&quot;middleroad&quot;&gt;{{cite news |title=They Took The Middle Road |work=[[St. Louis Post-Dispatch]] |date=7 March 2004 |page=E4 |id={{ProQuest|402424327}} |quote=James Timothy McCarver, baseball catcher; Lynn Nolan Ryan, baseball pitcher; Michael Sylvester Stallone, actor}}{{subscription required}}&lt;/ref&gt;

3552 &lt;ref name=&quot;slystallone&quot;&gt;Michael Sylvester Stallone

3553 *{{cite book | last=Aste | first=Mario | chapter=Movie Actors And Actresses | editor1=LaGumina, Salvatore J. | editor2=Cavaioli, Frank J. | editor3=Primeggia, Salvatore | editor4=Varacalli, Joseph A. | title=The Italian American Experience: An Encyclopedia | publisher=[[Garland Publishing]] | year=2000 | isbn=978-0-8153-0713-6 | url=https://books.google.com/books?id=3MpHvK-mAwYC | page=393 | quote=Sylvester Stallone, actor, director, and screenwriter, was born Michael Sylvester Stallone in 1946 in New York City | access-date=August 16, 2024 | archive-date=July 1, 2024 | archive-url=https://web.archive.org/web/20240701130229/https://books.google.com/books?id=3MpHvK-mAwYC | url-status=live }}

3554 *{{cite book |last1=Katz |first1=Ephraim |author1-link=Ephraim Katz |editor1-last=Klein |editor1-first=Fred |editor2-last=Nolen |editor2-first=Ronald Dean |title=The Film Encyclopedia |date=2001 |publisher=[[HarperResource]] |location=New York |isbn=0-06-273755-4 |page=1293 |edition=4th |url=https://archive.org/details/filmencyclopedia0004katz/page/n3/mode/2up |volume=4 |url-access=registration |quote=Stallone, Sylvester. Actor, director, screenwriter. Born Michael Sylvester Stallone, on July 6, 1946, in New York City }}

3555 *{{cite news |last1=Prost |first1=Marlene A. |title=Michael Sylvester Stallone |work=[[Philadelphia Inquirer]] |date=7 May 1987 |page=M4 |quote=Michael Sylvester Stallone, who was slight in stature, began working out and went out for football, track and other sports.}}

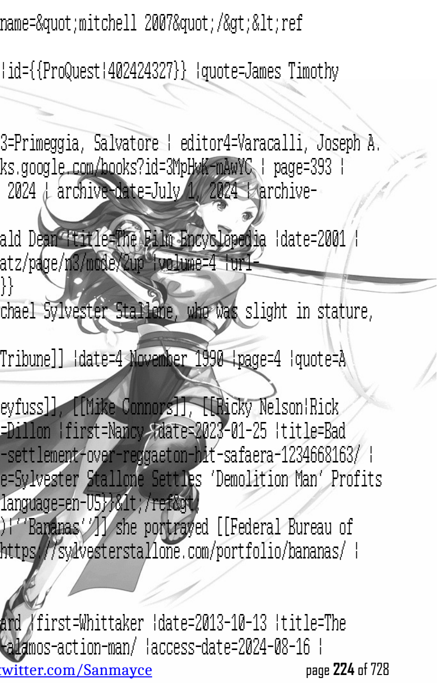
3556 *{{cite news |last1=Plaskin |first1=Glenn |title=Stallone's Long 'Rocky' Road Outcast Boy From N.Y. Is Still Fighting For Respect |work=[[Chicago Tribune]] |date=4 November 1990 |page=4 |quote=A botched birth, July 6, 1946 ... a physician delivered the baby, Michael Sylvester Stallone...}}&lt;/ref&gt;

3557 Johnson has represented high-profile clients such as [[Sylvester Stallone]], [[John Lennon]], [[Buddy Holly]], [[Michelle Phillips]], [[Richard Dreyfuss]], [[Mike Connors]], [[Ricky Nelson|Rick Nelson]], [[P. F. Sloan|P.F. Sloan]], members of [[Earth, Wind &amp; Fire]], [[Mitch Ryder]], and [[Lloyd Price]].&lt;ref&gt;{{Cite magazine |last=Dillon |first=Nancy |date=2023-01-25 |title=Bad Bunny Reaches Preliminary Settlement Over Reggaeton Hit 'Safaera' |url=https://www.rollingstone.com/music/music-news/bad-bunny-reaches-preliminary-settlement-over-reggaeton-hit-safaera-1234668163/ |access-date=2024-05-26 |magazine=Rolling Stone |language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite web |last=Maddaus |first=Gene |date=2019-05-08 |title=Sylvester Stallone Settles 'Demolition Man' Profits Dispute |url=https://variety.com/2019/biz/news/sylvester-stallone-demolition-man-settlement-1203208843/ |access-date=2024-05-26 |website=Variety |language=en-US}}&lt;/ref&gt;

3558 Fox moved to the big screen in the 1970s starring in films of the [[Blaxploitation|blaxploitation genre]]. In the 1971 comedy film [[Bananas (film)|''Bananas'']] she portrayed [[Federal Bureau of Investigation]] director [[J. Edgar Hoover]] disguised as a black woman.&lt;ref&gt;{{Cite web |title=Bananas (1971) {{!}} Sylvester Stallone |url=https://sylvesterstallone.com/portfolio/bananas/ |access-date=2024-08-21 |website=sylvesterstallone.com}}&lt;/ref&gt;

3559 |&lt;small&gt;[[Sylvester Stallone]]&lt;/small&gt;

3560 In addition to making movie montages for the Drafthouse, MacLennan was well-known for his mystery movie marathons.&lt;ref&gt;{{Cite web |last=Richard |first=Whittaker |date=2013-10-13 |title=The Alamo's Action Man: Programmer Greg MacLennan on the world's most ridiculous job |url=https://www.austinchronicle.com/daily/screens/2013-10-13/the-alamos-action-man/ |access-date=2024-08-16 |



website=Austin Chronicle }}&lt;/ref&gt; Most notable of these marathons included Cruise Control (featuring [[Tom Cruise]] movies), RussellMania (featuring [[Kurt Russell]] films), Baymaggedon (featuring [[Michael Bay]] films), The Stallone Zone (featuring [[Sylvester Stallone]] films), and CAGED (featuring [[Nicolas Cage]] films).&lt;ref&gt;{{Cite web ¦last=Travis ¦first=Ed ¦date=2014-08-04 ¦title=Cruise Control: A Trust Fall Into The Arms Of Tom Cruise &amp; Alamo Drafthouse's Greg MacLennan - Cinapse ¦url=https://cinapse.co/2014/08/cruise-control-a-trust-fall-into-the-arms-of-tom-cruise-alamo-drafthouses-greg-maclennan/ ¦access-date=2024-08-16 ¦website=Cinapse - Cinema Discovery and Discussion ¦language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦last=Whittaker ¦first=Richard ¦date=2017-05-03 ¦title=Get Ready for Russellmania II ¦url=https://www.austinchronicle.com/daily/screens/2017-05-03/get-ready-for-russellmania-ii/ ¦access-date=2024-08-16 ¦website=Austin Chronicle}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦last=Schillaci ¦first=Sophie ¦date=2013-04-18 ¦title=Michael Bay to Be Honored at Alamo Drafthouse 'Baymageddon' Event (Exclusive) ¦url=https://www.hollywoodreporter.com/news/general-news/michael-bay-alamo-drafthouse-honoring-442257/ ¦access-date=2024-08-16 ¦website=The Hollywood Reporter ¦language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦last=Whittaker ¦first=Richard ¦date=2013-08-24 ¦title=Rocky, Rambo and Rhinestones With Sly Stallone ¦url=https://www.austinchronicle.com/daily/screens/2013-08-24/rocky-rambo-and-rhinestones-with-sly-stallone/ ¦access-date=2024-06-16 ¦website=Austin Chronicle}}&lt;/ref&gt;

3561 In 2007, Jeff Ruby's Steakhouse in Louisville made national news when Jeff refused to serve [[O. J. Simpson]].&lt;ref&gt;{{cite news ¦last1=Harvey ¦first1=Scott ¦date=May 9, 2007 ¦title=Restaurant owner says O.J. Simpson just not welcome ¦url=https://www.wave3.com/story/6493883/restaurant-owner-says-oj-simpson-just-not-welcome/ ¦access-date=2024-07-13 ¦publisher=Wave 3}}&lt;/ref&gt; Also in 2016, he banned [[Donald Trump]] from his restaurant during a campaign visit due to insulting comments Trump made about Arizona Sen. [[John McCain]] and other [[POWs]], questioning their status as heroes.&lt;ref&gt;{{Cite web ¦last=Aldridge ¦first=Kevin S. ¦title=He kicked out O.J. Simpson, banned Donald Trump from his restaurant. But what about Sarah Sanders? ¦url=https://www.usatoday.com/story/opinion/nation-now/2018/06/28/sara-huckabee-sanders-got-ojed-civility-column/739697002/ ¦access-date=2024-07-13 ¦website=USA TODAY ¦language=en-US}}&lt;/ref&gt; In 2023, Ruby catered a special dinner for the [[Damar Hamlin]] family at the [[University of Cincinnati Academic Health Center¦University of Cincinnati Medical Center]].&lt;ref&gt;{{Cite web ¦last=Ogwude ¦first=Haadiza ¦title='It becomes personal for me': Restauranteur Jeff Ruby caters free dinner for Damar Hamlin's family ¦url=https://www.usatoday.com/story/sports/nfl/2023/01/04/jeff-ruby-caters-free-dinner-damar-hamlin-family/10991902002/ ¦access-date=2024-07-13 ¦website=USA TODAY ¦language=en-US}}&lt;/ref&gt; Various celebrities, including [[Drake (musician)¦Drake]] and [[Sylvester Stallone]], are frequent diners at Jeff Ruby's Steakhouse.&lt;ref&gt;{{Cite web ¦last=Hodges ¦first=Bebe ¦date=2024-03-02 ¦title='Rocky' actor Sylvester Stallone spotted at Jeff Ruby's Steakhouse ¦url=https://www.cincinnati.com/story/entertainment/2024/03/02/rocky-actor-sylvester-stallone-at-jeff-rubys-steakhouse/72821122007/ ¦access-date=2024-07-13 ¦website=The Enquirer ¦language=en-US}}&lt;/ref&gt;&lt;ref&gt;{{Cite web ¦last=Behrens ¦first=Cole ¦date=2024-02-23 ¦title=Here's where Drake stopped for dinner in Columbus after one of his shows ¦url=https://www.dispatch.com/story/entertainment/music/2024/02/23/drake-columbus-restaurant-after-shows-jeff-ruby-steakhouse/72710220007/ ¦access-date=2024-07-13 ¦website=The Columbus Dispatch ¦language=en-US}}&lt;/ref&gt;

3562 I've listed this article for peer review because I am hoping to eventually turn it into a featured article. ''Tulsa King'' is a fantastic crime and comedy drama starring the one and only Sylvester Stallone. Just this year I have expanded it from basically a stub and turned it into a well fleshed out good article. I nominated it for FA shortly after, but it was quickly closed and suggested that it go through GOCE and PR. The copy edit was just recently performed, so I'm hoping to work out any other issues in this review before I send it back to FAC. {{smiley}}

3563 '''Youssef Mansour''' ([[Arabic]]: يوسف منصور) is an Egyptian martial artist, actor, producer, and director best known for his [[martial arts movie¦martial arts films]].&lt;ref&gt;{{Cite web¦url=https://elcinema.com/en/person/1029757/¦title=Youssef Mansour - Actor Filmography، photos، Video¦website=elCinema.com}}&lt;/ref&gt;&lt;ref name=&quot;auto3&quot;&gt;{{Cite web¦url=http://www.mafhoum.com/press5/148S63.htm¦title=Cairo Times¦website=www.mafhoum.com}}&lt;/ref&gt;&lt;ref name=&quot;auto1&quot;&gt;{{Cite web¦url=https://www.masrawy.com/arts/zoom/details/2023/9/30/2474934/%D8%A8%D8%B9%D8%AF-15-%D8%B9%D8%A7%D9%85-%D8%A7-%D8%B9%D9%84%D9%89-%D9%88%D9%81%D8%A7%D8%AA%D9%87%D8%A7-%D9%8A%D9%88%D8%B3%D9%81-%D9%85%D9%86%D8%B5%D9%88%D8%B1-%D9%8A%D9%81%D8%AC%D8%B1-%D9%85%D9%81%D8%A7%D8%AC%D8%A3%D8%A9-%D8%B9%D9%86-%D8%B3%D9%88%D8%B2%D8%A7%D9%86-%D8%AA%D9%85%D9%8A%D9%85?amp¦title= بعد 15 عامًا على وفاتها.. يوسف منصور يفجر مفاجأة عن سوزان تميم¦website=www.masrawy.com}}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://cairoscene.com/buzz/stallone-star-in-youssef-mansour-new-movie¦title=Sylvester Stallone to Star in Egyptian film with Youssef Mansour¦website=CairoScene}}&lt;/ref&gt;&lt;ref&gt;{{Cite web¦url=https://scoopempire.com/american-film-chosen-guard-shot-egypt-featuring-youssef-mansour/¦title=American Film 'The Chosen Guard' to Be Shot in Egypt Featuring Youssef Mansour¦first=Scoop¦last=Team¦date=February 12, 2017}}&lt;/ref&gt;&lt;ref name=&quot;auto2&quot;&gt;{{Cite web¦url=https://stepfeed.com/hollywood-film-to-be-shot-in-egypt-with-an-egyptian-cast-0167¦title=Hollywood film to be shot in Egypt with an Egyptian cast¦first=Rayana¦last=Khalaf¦date=February 14, 2017¦website=StepFeed}}&lt;/ref&gt;&lt;ref&gt;https://al-ain.com/article/yousef-mansour-profile&lt;/ref&gt;&lt;ref&gt;https://m.elwatannews.com/news/details/4113954&lt;/ref&gt; Born in Cairo, he trained in the style of [[Kung-fu]] in China and America before embarking on a film career. He entered the [[Cinema of Egypt¦Egyptian film industry]] by chance when he met director Ibrahim Afifi and was offered his first role in the movie ''El Agooz Wil Baltagi'' (&quot;The Old Man and the Thug&quot;) in 1989. He became famous in the 1990s for starring in Egyptian films that relied on martial arts, such as ''Qabdet El Hilaly'' (&quot;El Hilaly's Fist&quot;) in 1991, ''El Shares'' (&quot;The Fierce&quot;) in 1992, and &quot;Desert Cat&quot; in 1995.&lt;ref name=&quot;auto2&quot;/&gt;&lt;ref name=&quot;auto1&quot;/&gt;

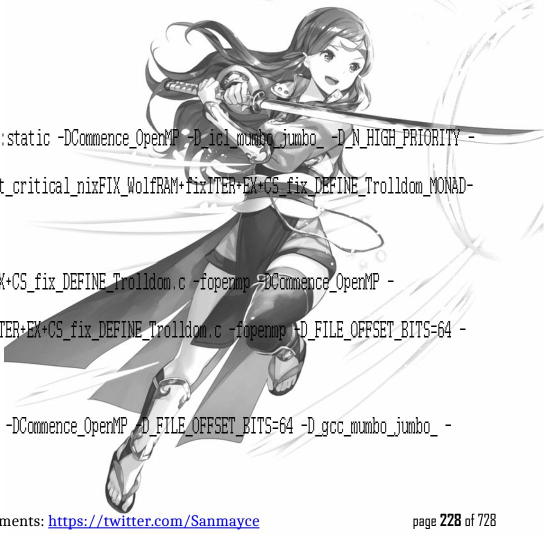
3564 The script for the film was written by Avinash Vijaykumar, who drew inspiration from two real-life incidents. One incident occurred in a village in [[North Karnataka]],&lt;ref name=&quot;:ZN&quot;&gt;{{cite news ¦title=My hero Kannada Movie releasing on august 30 {{!}} ನೈಜ ಘಟನೆ ಆಧರಿತ ಮೈ ಹೀರೋ ಚಿತ್ರ ಆಗಸ್ಟ್ 30ಕ್ಕೆ ಬಿಡುಗಡೆ {{!}} News in Kannada ¦url=https://zeenews.india.com/kannada/entertainment/my-hero-kannada-movie-releasing-on-august-30-236917/amp ¦work=zeenews.india.com ¦date=28 August 2024 ¦access-date=14 September 2024 ¦archive-date=15 September 2024 ¦archive-url=https://web.archive.org/web/20240915052938/https://zeenews.india.com/kannada/entertainment/my-hero-kannada-movie-releasing-on-august-30-236917/amp ¦url-status=live }}&lt;/ref&gt; where an [[Dalit¦oppressed community]] boy's hand was severed after he attempted to touch an idol during a religious procession. Another incident involved [[Sylvester Stallone]]'s visit to [[Varanasi]] for a [[Hinduism¦Hindu]] Tithi Shradh ritual in memory of his son, who passed away in 2012.&lt;ref name=&quot;:DH&quot; /&gt; These two incidents were incorporated into a single narrative.&lt;ref&gt;{{cite news ¦last1=Kotagunasi ¦first1=Manjunath B. ¦title=ದೇವರನ್ನು ಮುಟ್ಟಿದ ಬಾಲಕನಿಗೆ ಇದೆಂಥ ಶಿಕ್ಷೆ! ನೈಜ ಘಟನೆ ಆಧರಿತ ಮೈ ಹೀರೋ ಚಿತ್ರದಲ್ಲಿ ಹಾಲಿವುಡ್ ನಟ ಎರಿಕ್ ರಾಬರ್ಟ್ ¦url=https://kannada.hindustantimes.com/amp/entertainment/sandalwood-news-my-hero-kannada-movie-releasing-this-friday-august-29th-entertainment-news-in-kannada-mnk-181724855667957.html ¦work=Kannada Hindustan Times ¦date=28 August 2024 ¦language=kn ¦access-date=14 September 2024 ¦archive-date=15 September 2024 ¦archive-url=https://web.archive.org/web/20240915052802/https://kannada.hindustantimes.com/amp/entertainment/sandalwood-news-my-hero-kannada-movie-releasing-this-friday-august-29th-entertainment-news-in-kannada-mnk-181724855667957.html ¦url-status=live }}&lt;/ref&gt;&lt;ref&gt;{{cite news ¦title=My Hero: ನೈಜ ಘಟನೆ ಆಧಾರಿತ ಮೈ ಹೀರೋ ¦url=https://m.udayavani.com/article/my-hero-kannada-movie/1695297?utm=relatednews ¦work=[[Udayavani]] ¦date=30 August 2024 ¦access-date=15 September 2024 ¦archive-date=15 September 2024 ¦archive-url=https://web.archive.org/web/20240915052803/https://m.udayavani.com/article/my-hero-kannada-movie/1695297?utm=relatednews ¦url-status=live }}&lt;/ref&gt; Upon finishing the writing, Avinash met Djilali Rez Kallah at an institute,&lt;ref&gt;{{cite news ¦title=Djilali Rez-Kallah thrilled to debut in Sandalwood at a time when Indian movies are going global ¦url=https://timesofindia.indiatimes.com/entertainment/kannada/movies/news/djilali-rez-kallah-thrilled-to-debut-in-sandalwood-at-a-time-when-indian-movies-are-going-global/amp_articleshow/98178691.cms ¦work=The Times of India ¦date=24 February 2023 ¦access-date=14 September 2024 ¦archive-date=14 September 2024 ¦archive-url=https://web.archive.org/web/20240914071741/https://timesofindia.indiatimes.com/entertainment/kannada/movies/news/djilali-rez-kallah-thrilled-to-debut-in-sandalwood-at-a-time-when-indian-movies-are-going-global/amp_articleshow/98178691.cms ¦url-status=live }}&lt;/ref&gt; where he was chosen to portray the role of a retired Army officer.&lt;ref name=&quot;:TN1&quot;&gt;{{cite news ¦title=Sandalwood And Hollywood Actors Team Up In Kannada Film 'My Hero' ¦url=https://www.timesnownews.com/entertainment-news/kannada/sandalwood-and-hollywood-actors-team-up-in-kannada-film-my-hero-article-112445286/amp ¦work=Times Now ¦date=11 August 2024 ¦language=en ¦access-date=14 September 2024 ¦archive-date=14 September 2024 ¦archive-url=https://web.archive.org/web/20240914071721/https://www.timesnownews.com/entertainment-news/kannada/sandalwood-and-hollywood-actors-team-up-in-kannada-film-my-hero-article-112445286/amp ¦url-status=live }}&lt;/ref&gt; [[Eric Roberts]] was selected for a key role in the film, and his portions were filmed at his residence in [[Los Angeles]].&lt;ref&gt;{{cite news ¦title=Was incredible: Hollywood actor Eric Roberts on his Kannada debut ¦url=https://timesofindia.indiatimes.com/entertainment/kannada/movies/news/was-incredible-hollywood-actor-eric-roberts-on-his-kannada-debut/amp_articleshow/103472026.cms ¦work=The Times of India ¦date=8 September 2023 ¦access-date=14 September 2024 ¦archive-date=14 September 2024 ¦archive-url=https://web.archive.org/web/20240914071832/https://timesofindia.indiatimes.com/entertainment/kannada/movies/news/was-incredible-hollywood-actor-eric-roberts-on-his-kannada-debut/amp_articleshow/103472026.cms ¦url-status=live }}&lt;/ref&gt; ''[[Nammane Yuvarani]]'' fame Ankita Amar was chosen to play an NGO worker from [[Madhya Pradesh]].&lt;ref&gt;{{cite news ¦title=My Hero ಹುಡುಕಿಕೊಂಡ 'ನಮ್ಮನೆ ಯುವರಾಣಿ'; ಮಧ್ಯಪ್ರದೇಶದ ಮಹಿಳೆಯಾದ ಅಂಕಿತಾ ಅಮರ್ ಸಿನಿಮಾ ರಿಲೀಸ್ ಯಾವಾಗ? ¦url=https://vijaykarnataka.com/tv/news/nammane-yuvarani-serial-actress-ankita-amar-my-hero-kannada-movie/amp_articleshow/112670262.cms ¦work=[[Vijaya Karnataka]] ¦date=21 August 2024 ¦language=kn ¦access-date=14 September 2024 ¦archive-date=14 September 2024 ¦archive-url=https://web.archive.org/web/20240914071721/https://vijaykarnataka.com/tv/news/nammane-yuvarani-serial-actress-ankita-amar-my-hero-kannada-movie/amp_articleshow/112670262.cms ¦url-status=live }}&lt;/ref&gt;&lt;ref name=&quot;:ETV&quot; /&gt; Principal photography for the film began on 24 February 2023&lt;ref name=&quot;:CE1&quot;&gt;{{cite news ¦last1=Sharadhaa ¦first1=A. ¦title=Avinash Vijaykumar's My Hero marks the debut of Hollywood actor Djilali Rez Kallah ¦url=https://www.cinemaexpress.com/amp/story/kannada/interviews/2023/Feb/21/avinash-vijaykumars-my-heromarks-the-debut-of-hollywood-actor-djilali-rez-kallah-40188.html ¦work=Cinema Express ¦date=7 February 2023 ¦language=en ¦access-date=14 September 2024 ¦archive-date=14 September 2024 ¦archive-url=https://web.archive.org/web/20240914071735/https://www.cinemaexpress.com/amp/story/kannada/interviews/2023/Feb/21/avinash-vijaykumars-my-heromarks-the-debut-of-hollywood-actor-djilali-rez-kallah-40188.html ¦url-status=live }}&lt;/ref&gt; with [[muhurat shot]] pooja held in [[Basavanagudi]].&lt;ref&gt;{{cite news ¦title=National pride - Bangalore Mirror ¦url=https://bangaloremirror.indiatimes.com/entertainment/south-masala/national-pride/amp_articleshow/98284534.cms ¦work=bangaloremirror.indiatimes.com ¦date=28 February 2023 ¦access-date=14 September 2024 ¦archive-date=28 February 2023 ¦archive-url=https://web.archive.org/web/20230228052849/https://bangaloremirror.indiatimes.com/entertainment/south-masala/national-pride/amp_articleshow/98284534.cms ¦url-status=live }}&lt;/ref&gt; Major portions of the filming took place in [[Maheshwar]].&lt;ref name=&quot;:CE1&quot; /&gt;&lt;ref&gt;{{cite news ¦title=Bhopal's Shivam Pandey Featured In Kannada Film 'My Hero' With Hollywood Actors ¦url=https://www.freepressjournal.in/amp/bhopal/bhopals-shivam-pandey-featured-in-kannada-film-my-hero-with-hollywood-actors ¦work=[[The Free Press Journal]] ¦date=1 September 2024 ¦language=en ¦access-date=14 September 2024 ¦archive-date=9 September 2024 ¦archive-url=https://web.archive.org/web/20240909001006/https://www.freepressjournal.in/amp/bhopal/bhopals-shivam-pandey-featured-in-kannada-film-my-hero-with-hollywood-actors ¦url-status=live }}&lt;/ref&gt;&lt;ref name=&quot;:ETV&quot;&gt;{{cite news ¦title='ಮೈ ಹೀರೋ' ಚಿತ್ರದ ಮೂಲಕ ಕನ್ನಡಕ್ಕೆ ಬಂದ ಹಾಲಿವುಡ್ ನಟರು: ಟ್ರೇಲರ್ ನೋಡಿ - My Hero Trailer ¦url=https://www.etvbharat.com/amp/kn/!entertainment/hollywood-actors-came-to-kannada-film-industry-through-my-hero-watch-the-trailer-karnataka-news-kas24082102464 ¦work=ETV Bharat News ¦date=21 August 2024 ¦language=kn ¦access-date=14 September 2024 ¦archive-date=14 September 2024 ¦archive-url=https://web.archive.org/web/20240914071834/https://www.etvbharat.com/amp/kn/!entertainment/hollywood-actors-came-to-kannada-film-industry-through-my-hero-watch-the-trailer-karnataka-news-kas24082102464 ¦url-status=live }}&lt;/ref&gt; A few portions were shot in [[Chikmagalur]] and the United States.&lt;ref name=&quot;:CE1&quot; /&gt; The filming was completed within fifty days, including the fifteen days to finish the segments of the United States and thirty-five days for the India portions.&lt;ref name=&quot;:NIE2&quot;&gt;{{cite news ¦title='My Hero' is a culmination of my deep-seated love for storytelling, says director Avinash Vijaykumar ¦url=https://www.newindianexpress.com/amp/story/entertainment/kannada/2024/Aug/28/my-hero-is-a-culmination-of-my-deep-seated-love-for-storytelling-says-director-avinash-vijaykumar ¦work=The New Indian Express ¦date=12 August 2024 ¦language=en ¦access-date=14 September 2024 ¦archive-date=14 September 2024 ¦archive-url=https://web.archive.org/web/20240914071749/https://www.newindianexpress.com/amp/story/entertainment/kannada/2024/Aug/28/my-hero-is-a-culmination-of-my-deep-seated-love-for-storytelling-says-director-avinash-vijaykumar ¦url-status=live }}&lt;/ref&gt;&lt;ref name=&quot;:ZN&quot; /&gt; Forrest Chiras and Akula Lokesh Babu handled the cinematography for the United States portions,

```
while Venus Nagaraj Murthy handled the cinematography for the India portions.&lt;ref name=&quot;:NIE2&quot; /&gt; The editing was handled by Muthu Raj T.&lt;ref name=&quot;:NIE1&quot; /&gt;
According to ''[[Muscle and Fitness]]'' magazine, he considered [[Arnold Schwarzenegger]] and [[Sylvester Stallone]] as his basic source of inspiration.&lt;ref&gt;{{Cite web |date=14 September 2024 |title=World's 'most monstrous' bodybuilder Illia Yefimchyk dies of heart attack at 36 |url=https://indianexpress.com/article/trending/trending-globally/worlds-most-monstrous-bodybuilder-illia-yefimchyk-dies-of-heart-attack-at-36-9567052/ |access-date=14 September 2024 |website=The Indian Express |language=en}}&lt;/ref&gt;&lt;ref name=&quot;:0&quot;&gt;{{Cite news |date=13 September 2024 |title=Illia 'Golem' Yefimchyk, the world's 'most monstrous bodybuilder', died at 36; here's why he became famous |url=https://economictimes.indiatimes.com/news/international/us/illia-golem-yefimchyk-the-worlds-most-monstrous-bodybuilder-died-at-36-heres-why-he-became-famous/articleshow/113329355.cms?from=mdr |access-date=14 September 2024 |work=The Economic Times |issn=0013-0389}}&lt;/ref&gt; As a teenager, Yefimchyk weighed approximately {{convert|70|kg|lb}}, and was not able to perform [[pushup]]s. He adopted a strict training schedule and work ethic, including [[exercise physiology]].&lt;ref name=&quot;:0&quot; /&gt;
| [[Sylvester Stallone]]
| [[Sylvester Stallone]]
In 2023, ''[[Sylvester Stallone|Sly]]'' was chosen as the closing night film at the [[Toronto International Film Festival]].&lt;ref&gt;{{Cite web |last=Vlessing |first=Etan |date=2023-08-08 |title=Sylvester Stallone Netflix Doc 'Sly' to Close Toronto Film Festival |url=https://www.hollywoodreporter.com/movies/movie-news/sylvester-stallone-netflix-doc-toronto-1235558592/ |access-date=2024-10-04 |website=The Hollywood Reporter |language=en-US}}&lt;/ref&gt; [[Jazz Fest: A New Orleans Story]] won the Grammy Award for [[Grammy Award for Best Music Film|Best Music Film]].&lt;ref&gt;{{Cite web |last=McCarthy |first=Todd |date=2022-03-17 |title=SXSW Review: Frank Marshall &amp; Ryan Suffern's 'Jazz Fest: A New Orleans Story' |url=https://deadline.com/2022/03/jazz-fest-a-new-orleans-story-sxsw-review-frank-marshall-1234980917/ |access-date=2024-10-04 |website=Deadline |language=en-US}}&lt;/ref&gt;
```

The Kazahana full C sourcecode:

```
// 2024-Jan-29, changed '`' to '%' since the bash treats it as enclosure.
// 2024-Jan-24, added vectorized lowercasing.
// 2024-Jan-21, nasty bug (issue #2) was crushed, simply forgot to lowercase the haystack when Exact Searching in Insensitive mode, TO-DO: replace the scalar with vector LOWERcasing - it is in my old tools.
// 2024-Jan-10, nasty bug (issue #1) was crushed in 'if (cbTarget<777)' section of Trolldom, namely replacing 'unsigned long' with 'uint32_t' for:
// uint32_t ulHashPattern
// uint32_t ulHashTarget;
// uint32_t countSTATIC;
// uint32_t Quadruplet2nd;
// uint32_t Quadruplet3rd;
// uint32_t Quadruplet4th;
// uint32_t AdvanceHopperGrass;

// Vectorized lowercasing, according to ChatGPT, stupid-n-buggy:
/*
#include <stdio.h>
#include <emmintrin.h>
#include <ctype.h>

void vectorizedLowercase(char* data, size_t size) {
    // Make sure the size is a multiple of 16 (for SSE2)
    size_t alignedSize = size / 16 * 16;

    for (size_t i = 0; i < alignedSize; i += 16) {
        __m128i chunk = _mm_loadu_si128((__m128i*)(data + i));
        chunk = _mm_or_si128(_mm_and_si128(chunk, _mm_set1_epi8(0xDF)), _mm_set1_epi8(0x20));
        _mm_storeu_si128((__m128i*)(data + i), chunk);
    }

    // Process the remaining elements
    for (size_t i = alignedSize; i < size; ++i) {
        data[i] = (char)tolower(data[i]);
    }
}

int main() {
    char buffer[] = "Hello, World!";
    size_t size = sizeof(buffer) - 1;  // Exclude the null terminator

    vectorizedLowercase(buffer, size);

    // Print the result
    printf("%s\n", buffer);

    return 0;
}
*/

// Vectorized lowercasing, according to 'manatarka.h':
#include <immintrin.h>
#include <stdint.h> // Needed for uint32_t

void UCASE_XMM(uint64_t QWORDSsrc[], uint64_t QWORDSdst[], int64_t bytesto) {

       int64_t i = 0;
       __m128i maska;
       __m128i maskz;
       __m128i mask32;
       __m128i maskaz;
       __m128i r0;
       char *QWORDSs = (char *)QWORDSsrc;
       char *QWORDSd = (char *)QWORDSdst;

       maska = _mm_set1_epi8( 'a' );
       maskz = _mm_set1_epi8( 'z' );
       mask32 = _mm_set1_epi8( 32 );

       for( ; i < bytesto/16 * 16 ; i+=16 )
       {
             r0 = _mm_loadu_si128( ( __m128i * )&QWORDSs[ i ] );

             // maskaz contains 0x00 where character between 'a' and 'z', 0xff otherwise.
             maskaz = _mm_or_si128( _mm_cmplt_epi8( r0, maska ), _mm_cmpgt_epi8( r0, maskz ) );

             // Set the 6th bit to 0 only for lowercase characters.
             r0 = _mm_andnot_si128( _mm_andnot_si128( maskaz, mask32 ), r0 );
             _mm_storeu_si128( ( __m128i * )&QWORDSd[ i ], r0 );
       }
}

//void UCASE_XMM_inplace(uint64_t QWORDSsrc[], int64_t bytesto) {
void UCASE_XMM_inplace(char * QWORDSsrc, int64_t bytesto) {

       int64_t i = 0;
       __m128i maska;
       __m128i maskz;
       __m128i mask32;
       __m128i maskaz;
```

```
    __m128i r0;
    char *QWORDSs = (char *)QWORDSsrc;

    maska = _mm_set1_epi8( 'a' );
    maskz = _mm_set1_epi8( 'z' );
    mask32 = _mm_set1_epi8( 32 );

    for( ; i < bytesto/16 * 16 ; i+=16 )
    {
        r0 = _mm_loadu_si128( ( __m128i * )&QWORDSs[ i ] );

        // maskaz contains 0x00 where character between 'a' and 'z', 0xff otherwise.
        maskaz = _mm_or_si128( _mm_cmplt_epi8( r0, maska ), _mm_cmpgt_epi8( r0, maskz ) );

        // Set the 6th bit to 0 only for lowercase characters.
        r0 = _mm_andnot_si128( _mm_andnot_si128( maskaz, mask32 ), r0 );
        _mm_storeu_si128( ( __m128i * )&QWORDSs[ i ], r0 );
    }
}

#include <memory.h>
// Kazahana revision 1-++fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix_DEFINE_Trolldom_LineByLine, copyleft Kaze 2024-Jan-09.
// 2024-Jan-09:
// The buggy Exact search for whole block was commented out and replaced with LineByLine (as Wildcard is), thus uniformity is achieved.
// 2024-Jan-07:
// Some source formatting and CLANG warnings fixed, the 3rd thread is properly TABulated, was lazy to make the rest such.

// To-do, fix the exact search issue #1:
/*
[sanmayce@djudjeto5 Kazahana_ELF]$ ./Kazahana_Hexadecad_SSE42_64bit_CLANG-17.0.4.elf "xgrams" Kazahana.c 326
Kazahana: Dumped xgrams: 166

[sanmayce@djudjeto5 Kazahana_ELF]$ ./Kazahana_Hexadecad_SSE42_64bit_CLANG-17.0.4.elf "xgrams" Kazahana.c 726
Kazahana: Dumped xgrams: 168
*/

// 2019-May-21:
// Added some benchmark-related fragments, plus, Wolfram was replaced with Trolldom - the latter initializes the BMH table which slows the things down.

// Kazahana revision 1-++fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix_DEFINE, copyleft Kaze 2014-Dec-04.
// SUPERNASTY compiler-side mishavoc fixed (by lowering demands to static arrays with DEFINE 126+100 becoming 156), it appeared as random crashing/overwriting variables and general mayhem.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix, copyleft Kaze 2014-Nov-19.
// Fixed a stupid parsing bug causing FuzzyExhaustive to search in lines only up to 326 chars:
// MAXboth = MaxLineLength +1+1 +(167*WILDCARD_IP_flag*MaxLineLength); // Buggy line, fixed with next one in r. ...CS_fix
//    if (WILDCARD_IP_flag) {
//        MAXboth = MaxLineLength +1+1 +(167*WILDCARD_IP_flag*MaxLineLength);
//    } else {
//        MAXboth = MaxLineLength +1+1 +(167*EXHAUSTIVE_flag*MaxLineLength);
//    }
// Kazahana revision 1-++fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS, copyleft Kaze 2014-Mar-25.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX (MinGW ready, Linux ready), copyleft Sanmayce 2013-Dec-10.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX (MinGW ready, Linux ready), copyleft Sanmayce 2013-Dec-05.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+ (MinGW ready, Linux ready), copyleft Sanmayce 2013-Nov-30.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER (MinGW ready, Linux ready), copyleft Sanmayce 2013-Nov-29.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_WolfRAM+fix (MinGW ready, Linux ready), copyleft Sanmayce 2013-Nov-24.
// Stupid bug was crushed: 'unsigned int' became AGAIN 'int' as it was in Galadriel, simply forgot that it can be negative.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_WolfRAM+ (MinGW ready, Linux ready), copyleft Sanmayce 2013-Nov-21.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_WolfRAM (MinGW ready, Linux ready), copyleft Sanmayce 2013-Nov-15.
// Kazahana revision 1-++fix+nowait_critical_nixFIX_Bari (MinGW ready, Linux ready), copyleft Sanmayce 2013-Oct-23.

// Kazahana revision 1-++fix+nowait_critical_nixFIX (MinGW ready, Linux ready), copyleft Sanmayce 2013-Apr-07.
// Grr... a leftover/overlooked parsing bug was crushed.

// Kazahana revision 1-++fix+nowait_critical_nix (MinGW ready, Linux ready), copyleft Sanmayce 2013-Feb-24.
// TO-DO in r.1: recursive calls to be simulated with my own stack.
// In this revision 16 threads are enforced.
// If you want to help me to port it to *nix your name will appear as contributor in the credit part.
// Please give me a buzz (sanmayce@sanmayce.com) if you find faster implementation.
// Special thanks go to Igor Pavlov, VIVA.
// Enfun!

// How to compile under Windows using Intel compiler:
//icl /O3 /arch:SSE4.1 /MT Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_Trolldom.c /FAcs /FeKazahana_r1-+
    +fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_Trolldom_HEXADECAD-Threads_IntelV19_SSE41_64bit /Qopenmp /Qopenmp-link:static -DCommence_OpenMP -D_icl_mumbo_jumbo_ -D_N_HIGH_PRIORITY -
    D_WIN32_ENVIRONMENT_
//icl /O3 /arch:SSE4.1 /MT Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_Trolldom.c /FeKazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_Trolldom_MONAD-
    Thread_IntelV19_SSE41_64bit -D_icl_mumbo_jumbo_ -D_N_HIGH_PRIORITY -D_WIN32_ENVIRONMENT_

// How to compile under Windows using MinGW:
//gcc -O3 -m32 -funroll-loops -static -o Kazahana_Hexadecad_GCC_730_SSE41_32bit Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_Trolldom.c -fopenmp -DCommence_OpenMP -
    D_FILE_OFFSET_BITS=64 -D_gcc_mumbo_jumbo_ -D_N_HIGH_PRIORITY -D_WIN32_ENVIRONMENT_
//gcc -O3 -m32 -msse4.1 -funroll-loops -static -o Kazahana_Monad_GCC_730_SSE41_32bit Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_Trolldom.c -fopenmp -D_FILE_OFFSET_BITS=64 -
    D_gcc_mumbo_jumbo_ -D_N_HIGH_PRIORITY -D_WIN32_ENVIRONMENT_

// How to compile under *nix:
//gcc -O3 -funroll-loops -static -o kazahana Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_Trolldom.c -fopenmp -DCommence_OpenMP -D_FILE_OFFSET_BITS=64 -D_gcc_mumbo_jumbo_ -
    D_N_HIGH_PRIORITY -D_POSIX_ENVIRONMENT_

// Change accordingly from command line:
```



```
//#define _icl_mumbo_jumbo_
//#define _gcc_mumbo_jumbo_

// Change accordingly from command line:
//#define _WIN32_ENVIRONMENT_
//#define _POSIX_ENVIRONMENT_

#define _WildFastKaze_

// If you comment next, then the light-weight and faster on small haystacks 'Railgun_Quadruplet_7' will take over:
#define RG7Gulliver
// In fact Bari replaces Gulliver.

// How much MB the master-buffer will be? My tests show that 7 is a very good (but not excellent) value on my 4MB cache T7500.
// 11, 14, 19 are good values as they are one less than L3 cache of fast CPUs.
// Since Wikipedia has got some very long lines, 7 is also the minimal one if you want to search in her.
//#define MasterBuffer 7

/*
hatsutoukou : first (written) contribution
hatsuyuki : first snow (of season)
hatsuyume : year's first dream

fubuki : snow storm

amenochiyuki : rain then snow

fubon : uncommon, outstanding
fubuki : snow storm
fubun : unwritten, illiterate, uneducated

hyouden : field of eternal snow

koyuki : light snow
ooyuki : heavy snow

setsuzou : snow sculpture

shinshin : sound of heavy snow-fall
shinshin : mind body

yukionna : snow woman, fairy
*/

// [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Modified Igor Pavlov's wildcard.cpp from 7zip package wildcard code [

#ifndef NULL
#ifdef __cplusplus
#define NULL 0
#else
#define NULL ((void*)0)
#endif
#endif

#ifndef FALSE
#define FALSE 0
#endif
#ifndef TRUE
#define TRUE 1
#endif

#ifndef false
#define false 0
#endif
#ifndef true
#define true 1
#endif

typedef unsigned char char_t;
typedef char_t *string;
char TAGfree[10] = "*@#^$|~%.";
char TAGfreeFAST[3] = "&+"; // '&' stands for "standard" '*', '+' stands for "standard" '?'.
typedef unsigned char boolean;

#if defined(_icl_mumbo_jumbo_)
// GetRDTSC() taken from strchr.com
#if defined(_M_IX86)
unsigned long long __forceinline GetRDTSC(void) {
   __asm {
      ; Flush the pipeline
      XOR eax, eax
      CPUID
      ; Get RDTSC counter in edx:eax
      RDTSC
   }
}
#elif defined(_M_X64)
unsigned long long __forceinline GetRDTSC(void) {
```

```
        return __rdtsc();
}
#else
unsigned long long __forceinline GetRDTSC(void) {
        return GetTickCount();
}
#endif
#endif

#if defined(_gcc_mumbo_jumbo_)

#endif

#define KAZE_tolower(c) ( (((c) >= 'A') && ((c) <= 'Z')) ? ((c) - 'A' + 'a') : (c) )
#define KAZE_toupper(c) ( (((c) >= 'a') && ((c) <= 'z')) ? ((c) - 'a' + 'A') : (c) )

int maskGLOBALlen;
int nameGLOBALlen1;
int nameGLOBALlen2;
int nameGLOBALlen3;
int nameGLOBALlen4;
int nameGLOBALlen5;
int nameGLOBALlen6;
int nameGLOBALlen7;
int nameGLOBALlen8;
int nameGLOBALlen9;
int nameGLOBALlen0;
int nameGLOBALlena;
int nameGLOBALlenb;
int nameGLOBALlenc;
int nameGLOBALlend;
int nameGLOBALlene;
int nameGLOBALlenf;

int CaseSensitiveWildcardMatching_flag; // ZERO for INSENSITIVE, NONZERO for SENSITIVE

//long VIVA_IgorPavlov_invocations_global_counter = 0;
//long WildGLOBALhits = 0;

#include <stdint.h> // Needed for uint32_t
//typedef unsigned char uint8_t;
//typedef unsigned short uint16_t;
//typedef unsigned int uint32_t;

#ifdef _N_HIGH_PRIORITY
// https://msdn.microsoft.com/en-us/library/windows/desktop/ms686219.aspx
#include <stdio.h>
#include <windows.h>
#include <tchar.h>
#endif

// For short needles, and mainly haystacks, 'Doublet' is quite effective. Consider it or 'Quadruplet'.
// Fixed version from 2012-Feb-27.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Doublet (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        register uint32_t ulHashPattern;
        uint32_t ulHashTarget, count, countSTATIC;

        if (cbPattern > cbTarget) return(NULL);

        countSTATIC = cbPattern-2;

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = (*(uint16_t *)(pbPattern));

        for ( ;; ) {
              if ( ulHashPattern == (*(uint16_t *)(pbTarget-cbPattern)) ) {
                            count = countSTATIC;
                            while ( count && *(char *)(pbPattern+2+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+2+(countSTATIC-count)) ) {
                                          count--;
                            }
                            if ( count == 0 ) return((pbTarget-cbPattern));
              }
              pbTarget++;
              if (pbTarget > pbTargetMax) return(NULL);
        }
}

#define NeedleThreshold2vs4swampLITE 9+10 // Should be bigger than 9. BMH2 works up to this value (inclusive), if bigger then BMH4 takes over. Should be <=255 otherwise the 0|1 BMH2 should be used.
char * Railgun_Trolldom (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        uint32_t ulHashPattern;
        signed long count;

        unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
        unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
```

```
       uint32_t i, Gulliver;

       uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
       uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
       uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

       if (cbPattern > cbTarget) return(NULL);

       if ( cbPattern<4 ) {
             // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
             // ...
             pbTarget = pbTarget+cbPattern;
             ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
             if ( cbPattern==3 ) {
                          for ( ;; ) {
                                       if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                    if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                       }
                                       if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                    pbTarget++;
                                                    if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                       }
                                       pbTarget++;
                                       if (pbTarget > pbTargetMax) return(NULL);
                          }
             } else {
             }
             for ( ;; ) {
                          if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                          if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                          pbTarget++;
                          if (pbTarget > pbTargetMax) return(NULL);
             }
       } else { //if ( cbPattern<4 )
             if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//          }

         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
               if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
```

```
        AdvanceHopperGrass++;
        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
     }
  }
    }

    AdvanceHopperGrass++;

    pbTarget = pbTarget + AdvanceHopperGrass;
     if (pbTarget > pbTargetMax)
       return(NULL);
  }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0¦1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0¦1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//  } else { //if ( cbPattern<4 )
//      if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] ¦
    (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                        if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
    == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
//      } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

// Slower than Swampshine's simple 0¦1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
```

```
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
    cbPattern = PRIMALlengthCANDIDATE;
    pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
            // BMH order 2, needle should be >=4:
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
            for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
            i=0;
            while (i <= cbTarget-cbPattern) {
                Gulliver = 1; // 'Gulliver' is the skip
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                            count = cbPattern-4+1;
                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                count = count-4;
                            if ( count <= 0 ) return(pbTarget+i);
                        }
                    }
                } else Gulliver = cbPattern-(2-1);
                i = i + Gulliver;
                //GlobalI++; // Comment it, it is only for stats.
            }
            return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
            // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
            // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
            // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
            // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
            //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
            for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
            i=0;
            while (i <= cbTarget-cbPattern) {
                Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
            count = PRIMALlengthCANDIDATE-4+1;
            while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                count = count-4;
            if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                    }
                    Gulliver = 1;
                } else
                    Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                }
                i = i + Gulliver;
                //GlobalI++; // Comment it, it is only for stats.
            }
            return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
```

```
So the result on Core 2 Q9550s @2.83GHz:
--------------------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                         ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦------------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                                  ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don        ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                   ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe      ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦------------------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence         ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                    ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦------------------------------------------------------------------------------------------¦-----------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf                ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦------------------------------------------------------------------------------------------¦-----------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                           ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                           for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                           // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40         add eax, 64
//  00142 66 0f 7f 44 14
//        60               movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00   cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70               movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00      movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00      movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2            mov edx, eax
//  00167 72 d6            jb .B5.21
                           //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                           // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00               mov r8d, 65536
//  0012d 44 8b 26         mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff   call _intel_fast_memset
                           // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                           // we must switch to 0¦1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
       it is okay to use 'memset'. !
                           for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                           i=0;
                           while (i <= cbTarget-cbPattern) {
                                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
       (Order-1) chars i.e. one char for Order 2
                                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                     if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                  count = cbPattern-4+1;
                                                                  while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                               count = count-4;
                                                                  if ( count <= 0 ) return(pbTarget+i);
                                                     }
                                                     Gulliver = 1;
                                        } else
                                                     Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                        }
                                        i = i + Gulliver;
                                        //GlobalI++; // Comment it, it is only for stats.
                           }
                           return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
```

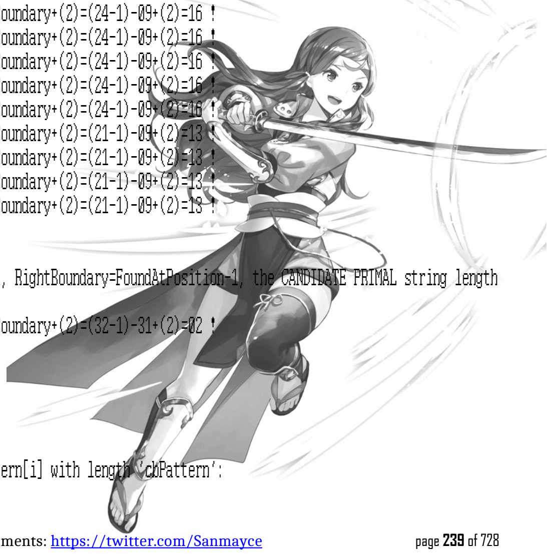
```
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
```

```
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
        pushq %r15
        .seh_pushreg    %r15
        movl  $65592, %eax
        pushq %r14
        .seh_pushreg    %r14
        pushq %r13
        .seh_pushreg    %r13
        pushq %r12
        .seh_pushreg    %r12
        pushq %rbp
        .seh_pushreg    %rbp
        pushq %rdi
        .seh_pushreg    %rdi
        pushq %rsi
        .seh_pushreg    %rsi
        pushq %rbx
        .seh_pushreg    %rbx
        call  ___chkstk_ms
        subq  %rax, %rsp
        .seh_stackalloc    65592
        .seh_endprologue
        cmpl  %r9d, %r8d
        movq  %rcx, %rbx
        movq  %rdx, %rdi
        movl  %r8d, %r12d
        movl  %r9d, %esi
        jb    .L118
        movl  (%rdx), %ebp
        leal  -1(%r9), %edx
        movl  $65536, %r8d
        leaq  48(%rsp), %rcx
        movzbl%dl, %edx
        call  memset
        movl  %esi, %r11d
        subl  $1, %r11d
        je    .L119
```

```
        xorl  %eax, %eax
        .p2align 4,,10
.L113:
        movzwl(%rdi,%rax), %edx
        movb  %al, 48(%rsp,%rdx)
        addq  $1, %rax
        cmpl  %eax, %r11d
        ja    .L113
.L112:
        leal  -4(%rsi), %r9d
        movl  %r12d, %r8d
        xorl  %edx, %edx
        leal  -3(%rsi), %eax
        shrl  $2, %r9d
        subl  %esi, %r8d
        leal  -2(%rsi), %r10d
        movslq%eax, %r14
        negq  %r9
        movl  %eax, 44(%rsp)
        leaq  -1(%r14), %r15
        salq  $2, %r9
        leaq  (%rdi,%r14), %r13
        jmp   .L117
        .p2align 4,,10
.L130:
        movl  %r10d, %eax
        subl  %ecx, %eax
        cmpl  %r10d, %ecx
        je    .L129
.L114:
        addl  %eax, %edx
        cmpl  %r8d, %edx
        ja    .L118
.L117:
        leal  (%rdx,%r10), %eax
        movzwl(%rbx,%rax), %eax
        movzbl48(%rsp,%rax), %ecx
        cmpl  %r11d, %ecx
        jne   .L130
        movl  %r11d, %eax
        addl  %eax, %edx
        cmpl  %r8d, %edx
        jbe   .L117
.L118:
        xorl  %eax, %eax
        jmp   .L128
        .p2align 4,,10
.L129:
        movl  %edx, %ecx
        movl  $1, %eax
        leaq  (%rbx,%rcx), %r12
        cmpl  (%r12), %ebp
        jne   .L114
        movl  44(%rsp), %esi
        testl %esi, %esi
        jle   .L124
        movl  (%r12,%r15), %esi
        cmpl  %esi, (%rdi,%r15)
        jne   .L114
        addq  %r14, %rcx
        xorl  %eax, %eax
        addq  %rbx, %rcx
        jmp   .L116
        .p2align 4,,10
.L132:
        movl  -5(%r13,%rax), %esi
        subq  $4, %rax
        cmpl  -1(%rcx,%rax), %esi
        jne   .L131
.L116:
        cmpq  %rax, %r9
        jne   .L132
.L124:
        movq  %r12, %rax
.L128:
        addq  $65592, %rsp
        popq  %rbx
        popq  %rsi
        popq  %rdi
        popq  %rbp
        popq  %r12
        popq  %r13
        popq  %r14
        popq  %r15
        ret
        .p2align 4,,10
.L131:
        movl  $1, %eax
        jmp   .L114
```

```
.L119:
       xorl  %r11d, %r11d
       jmp   .L112
*/

} //if (cbTarget<777)

             } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv             PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
       FoundAtPosition = cbPattern - ((4)-1) + 1;
       PRIMALpositionCANDIDATE=i;
       while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
             j = PRIMALpositionCANDIDATE + 1;
             while ( j <= (FoundAtPosition-1) ) {
                          if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                          j++;
             }
             PRIMALpositionCANDIDATE++;
       }
       PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
       if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
       if (cbPattern-i+1 <= PRIMALlength) break;
       if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv             PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                    PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                  PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                      PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                    PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                         PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                        PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                      PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                      PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                     PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;
```

```
                        00000000011111111112222222222333
                        12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
```

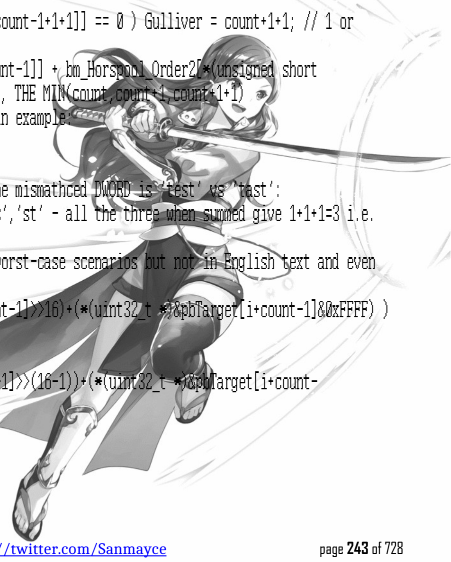


```
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
```

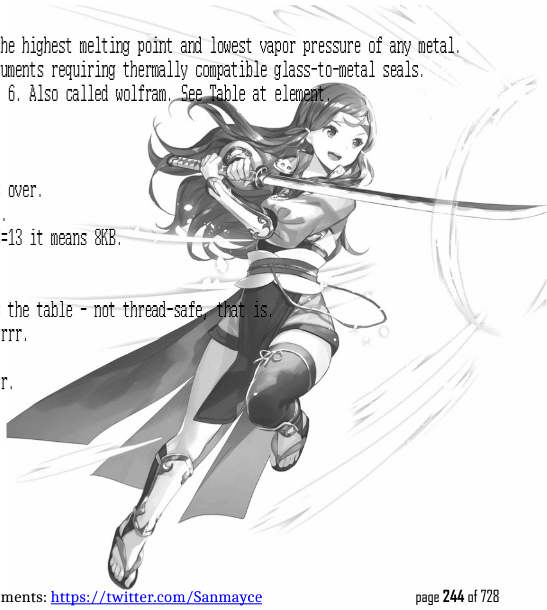
```
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00              mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
```

```
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
    #1
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
    the middle DWORD is 'bc'.
                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // Above line is replaced by next one with better hashing:
                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
    for ASCII texts with no symbols in range 128-255.
                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                // T           = 01010100
                                // TT          = 01010100 01010100
                                // TTTT        = 01010100 01010100 01010100 01010100
                                // TTTT>>16    = 00000000 00000000 01010100 01010100
                                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
    0 for English texts & 'ACGT' data.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // 'Maximus' uses branched 'if', again.
                                if ( \
                                ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
    1 )] ) == 0 \
                                || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
    1 )] ) == 0 \
                                ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                       |
//     DWORD #3                                                                     |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!  |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                    |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!     |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                    |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                       |
//     DWORD #3                                                                     |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!  |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                    |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!     |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                    |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                        // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
```

```
     'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                               //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                               //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                               //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                               //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                               //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                               //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                               //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                               //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                         count = cbPattern-4+1;
                                                         // Below comparison is UNIdirectional:
                                                         while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                  count = count-4;

     if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
     // The line below is BUGGY:
     //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
     // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
     //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
     if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
          if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
     below:
                   count = PRIMALlengthCANDIDATE-4+1;
                   while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                            count = count-4;
                   if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
          }
     }
}
     } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                         if ( count <= 0 ) return(pbTarget+i);
     }


                                                         // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                         // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                                       for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                                if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                                if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
     = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
     *)&pbTarget[i] == ulHashPattern)
//                                                       }
//                                                       if ( count <= 0 ) return(pbTarget+i);
                                                                  // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                                  //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it

     should

                                                                  //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,

     as it should

                                                                  //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or

     bigger, as it should
                                                         //       if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
     *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                                  // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                                  // Needle:    fastest tax
                                                                  // Window: ...fastast tax...
                                                                  // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                                  // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.

     Gulliver remains 1.

                                                                  // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even

     not in ACGT data, that's why I commented it in original 'Shockeroo'.

                                                                  //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )

     & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should

                                                                  // Above line is replaced by next one with better hashing:
//                                                                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
     1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                         // Order 4 ]
                                               }
                                      }
                            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                            i = i + Gulliver;
```



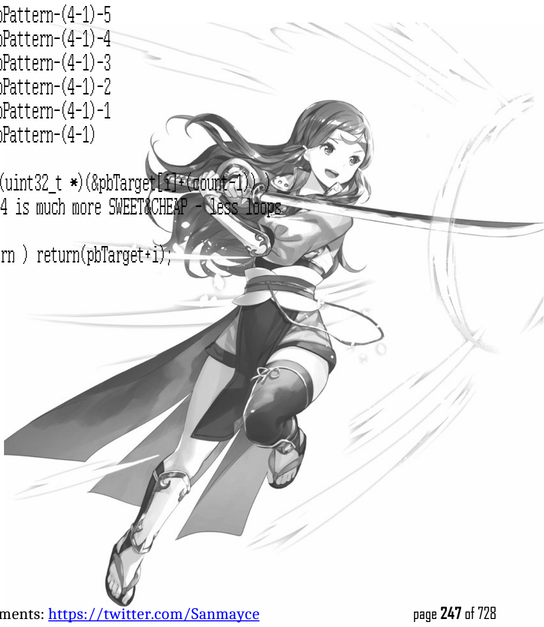
```
                        //GlobalI++; // Comment it, it is only for stats.
                    }
                    return(NULL);

            } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
            } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } //if ( cbPattern<4 )
}

/*
int main( void )
{
   DWORD dwError, dwPriClass;

   if(!SetPriorityClass(GetCurrentProcess(), REALTIME_PRIORITY_CLASS))
   {
      _tprintf(TEXT("Already REALTIME_PRIORITY\n"));
      goto Cleanup;
   }

   // Display priority class

   dwPriClass = GetPriorityClass(GetCurrentProcess());

   _tprintf(TEXT("Current priority class is 0x%x\n"), dwPriClass);

   if (dwPriClass==0x00000100) printf("Current priority class is REALTIME_PRIORITY_CLASS.\n");


Cleanup:
   // Clean up
   ;
return 0;
}

// IDLE_PRIORITY_CLASS
// 0x00000040
// Process whose threads run only when the system is idle. The threads of the process are preempted by the threads of any process running in a higher priority class. An example is a screen saver. The idle-priority class is inherited by child processes.

// NORMAL_PRIORITY_CLASS
// 0x00000020
// Process with no special scheduling needs.

// HIGH_PRIORITY_CLASS
// 0x00000080
// Process that performs time-critical tasks that must be executed immediately. The threads of the process preempt the threads of normal or idle priority class processes. An example is the Task List, which must respond quickly when called by the user, regardless of the load on the operating system. Use extreme care when using the high-priority class, because a high-priority class application can use nearly all available CPU time.

// REALTIME_PRIORITY_CLASS
// 0x00000100
// Process that has the highest possible priority. The threads of the process preempt the threads of all other processes, including operating system processes performing important tasks. For example, a real-time process that executes for more than a very brief interval can cause disk caches not to flush or cause the mouse to be unresponsive.
*/

#define ASIZE 256
// For speed up next 3 arrays are global:
unsigned int bm_bc[256]; //BMH needed
unsigned int bm_bc2nd[256]; //BMS needed

// Railgun_Sekireigan_Wolfram, copyleft 2013-Nov-11, Kaze.
// Do you know what is really COOL?
// Wolfram, if you ask me, with melting point of 3,410 Celsius.
// tungsten
// n. Symbol W
// A hard, brittle, corrosion-resistant, gray to white metallic element extracted from wolframite, scheelite, and other minerals, having the highest melting point and lowest vapor pressure of any metal.
// Tungsten and its alloys are used in high-temperature structural materials; in electrical elements, notably lamp filaments; and in instruments requiring thermally compatible glass-to-metal seals.
// Atomic number 74; atomic weight 183.84; melting point 3,410°C; boiling point 5,900°C; specific gravity 19.3 (20°C); valence 2, 3, 4, 5, 6. Also called wolfram. See Table at element.
// /Heritage/
#define _rotl_KAZE(x, n) (((x) << (n)) | ((x) >> (32-(n))))
#define HaystackThresholdSekireiTchittoGritto 961 // Quadruplet works up to this value, if bigger then BMH2 takes over.
#define NeedleThreshold2vs4TchittoGritto 22 // Should be bigger than 8. BMH2 works up to this value (inclusive), if bigger then BMH4 takes over.
#define NeedleThresholdBIGSekireiTchittoGritto 12+700 // Should be bigger than 'HasherezadeOrder'. BMH2 works up to this value (inclusive).
#define HashTableSizeSekireiTchittoGritto 17-1 // In fact the real size is -3, because it is BITwise, when 17-3=14 it means 16KB, (17-1)-3=13 it means 8KB.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.

// The line below was commented 2019-May-17, it is needed for Wolfram and old ones - it doesn't work for Trolldom since the latter changes the table - not thread-safe, that is.
//unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
// The line below was uncommented 2019-May-21, for speeding up:
unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];

// Commented 2019-May-17 [[[[[[[[
/*
char * Railgun_Sekireigan_Wolfram_1 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        register uint32_t ulHashPattern;
```



```
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
```

```
                    i=0;
                    //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                    //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                    //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                    for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                    while (i <= cbTarget-cbPattern) {
                            Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                            // Order 4 [
                                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                            count = cbPattern-4+1;
                                            //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                    count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                            if ( count <= 0 ) {
                                                    return(pbTarget+i);
                                            }
                                            //if ( count <= 0 ) {
                                            //              if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                            //}
                                            //else {
                                            //              if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                            //}
                                            // Order 4 ]
                                            }
                                    } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                            } else Gulliver = cbPattern-(2-1);
                            i = i + Gulliver;
                            //GlobalI++; // Comment it, it is only for stats.
                    }
                    return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                    //countSTATIC = cbPattern-2-2;
                    ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                    //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                    i=0;
                    //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                    //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                    //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
                    // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //"fast"
                    //"aste"
                    //"stes"
                    //"test"
                    //"est "
                    //"st f"
                    //"t fo"
                    //" fox"
                    //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
                    for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                    while (i <= cbTarget-cbPattern) {
                            Gulliver = 1;
                            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                                    if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                                    // Order 4 [
                                            // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                            // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
```

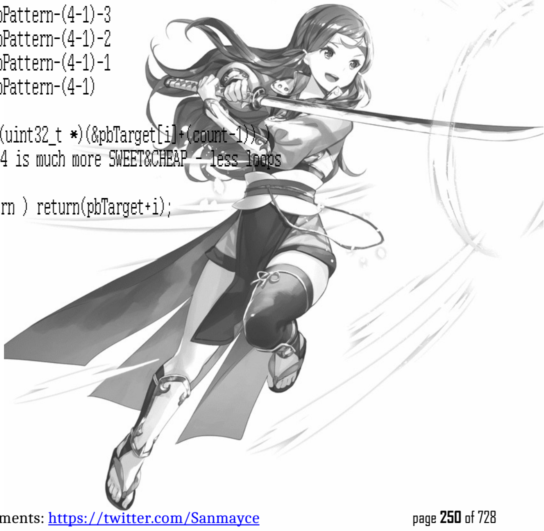
```
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //  if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
  ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
  } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

  } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
  cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
  'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        // Order 4 ]
                }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        } // while (i <= cbTarget-cbPattern)
        return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
  } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
    } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
  } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_2 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
```



```
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
```

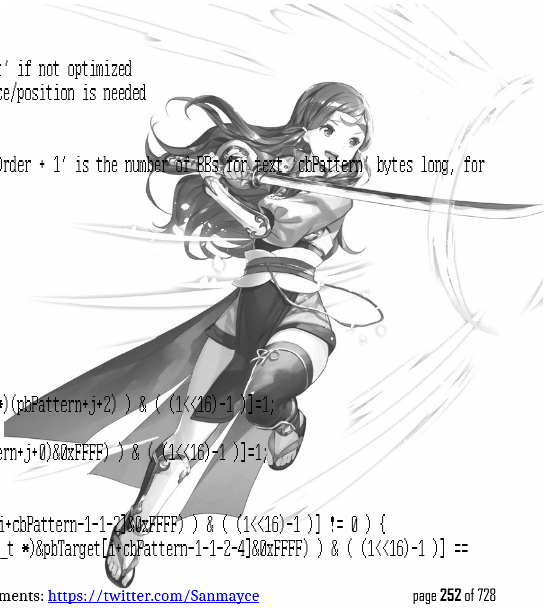
```
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
                for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
```

```
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //  if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        // Order 4 ]
                }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        } // while (i <= cbTarget-cbPattern)
        return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
      } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
```

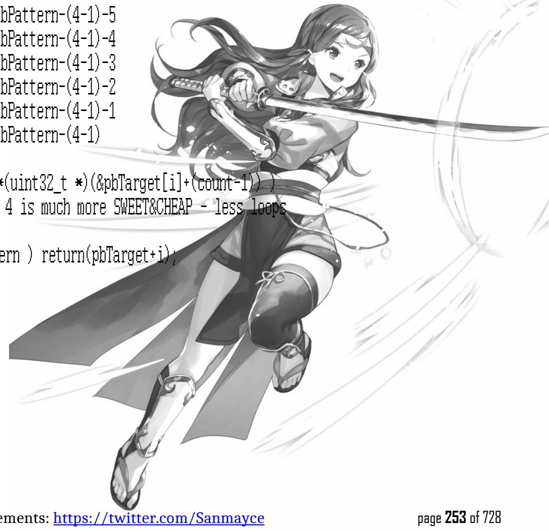


```
char * Railgun_Sekireigan_Wolfram_3 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, ‘char’ limits patterns to 255, if ‘long’ then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
            for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                    if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                }
                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                    pbTarget++;
                    if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                }
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
            }
        } else {
        }
        for ( ;; ) {
            if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
            if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
            pbTarget++;
            if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don’t know how exactly), it ensures (at least must) better performance than ‘Boyer_Moore_Horspool’.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
            AdvanceHopperGrass = 0;
            ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
            if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but ‘AdvanceHopperGrass’ must be calculated - it has a higher priority.
                count = cbPattern-1;
                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                    if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                    count--;
                }
                if ( count == 0) return((pbTarget-cbPattern));
            } else { // The goal here: to avoid memory accesses by stressing the registers.
                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                    AdvanceHopperGrass++;
                    if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                        AdvanceHopperGrass++;
                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                    }
                }
            }
            AdvanceHopperGrass++;
            pbTarget = pbTarget + AdvanceHopperGrass;
            if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
```

```
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
                for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
```

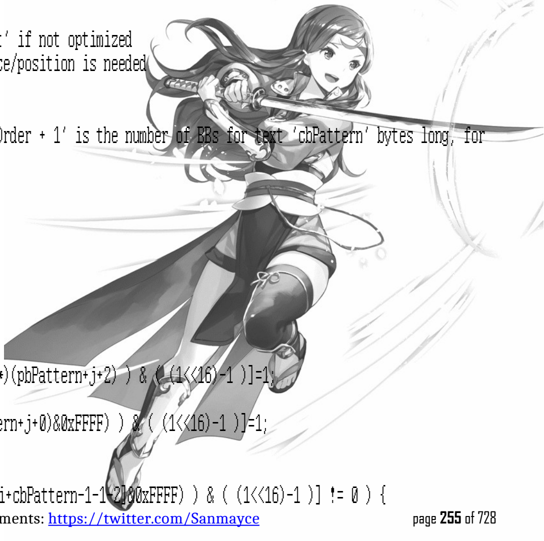


```
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
  'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //          if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
  ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
  } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

  } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
  cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )    Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
  'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        // Order 4 ]
                }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        } // while (i <= cbTarget-cbPattern)
        return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
  } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
    } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
```

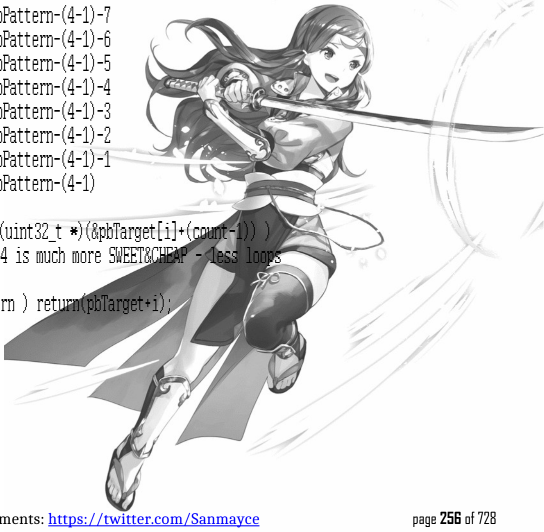


```
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_4 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {
```

```
    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
            //countSTATIC = cbPattern-2-2;
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
            i=0;
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
            //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
            for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
            while (i <= cbTarget-cbPattern) {
                Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            return(pbTarget+i);
                        }
                        //if ( count <= 0 ) {
                        //          if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        //}
                        //else {
                        //          if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                        }
                    } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                } else Gulliver = cbPattern-(2-1);
                i = i + Gulliver;
                //GlobalI++; // Comment it, it is only for stats.
            }
            return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
            //countSTATIC = cbPattern-2-2;
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
            i=0;
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
            //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
            // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
            //"fast"
            //"aste"
            //"stes"
            //"test"
            //"est "
            //"st f"
            //"t fo"
            //" fox"
            //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
            for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
            while (i <= cbTarget-cbPattern) {
                Gulliver = 1;
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
```



```
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4])>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] == 
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1; 
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) & 
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals 
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1; 
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        // Order 4 ]
                }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        } // while (i <= cbTarget-cbPattern)
        return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
```



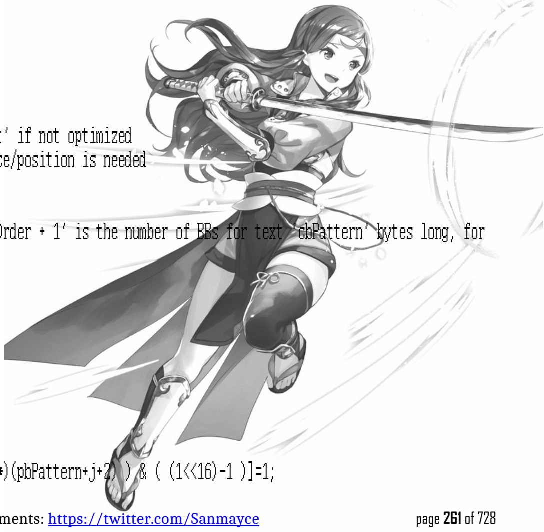
```
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_5 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }
```

```
    } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
  if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

  // BMH order 2:
  if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
        i=0;
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
        for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
  // Few thoughts regarding an excellent Skip Performance etude:
  // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
  // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
  // The code is like:
  // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
  // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
  // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
  // {
  // ...
  // }
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
  Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                    // Order 4 [
                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
  and Order=4 we have BBs = 11-4+1=8:
                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                    count = cbPattern-4+1;
                    //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                    while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                    if ( count <= 0 ) {
                        return(pbTarget+i);
                    }
                    //if ( count <= 0 ) {
                    //      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                    //}
                    //else {
                    //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                    //}
                    // Order 4 ]
                    }
                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
  // BMH order 4, needle should be >=8:
  } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
  example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
        //"st f"
        //"t fo"
        //" fox"
        //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
        for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        while (i <= cbTarget-cbPattern) {
```

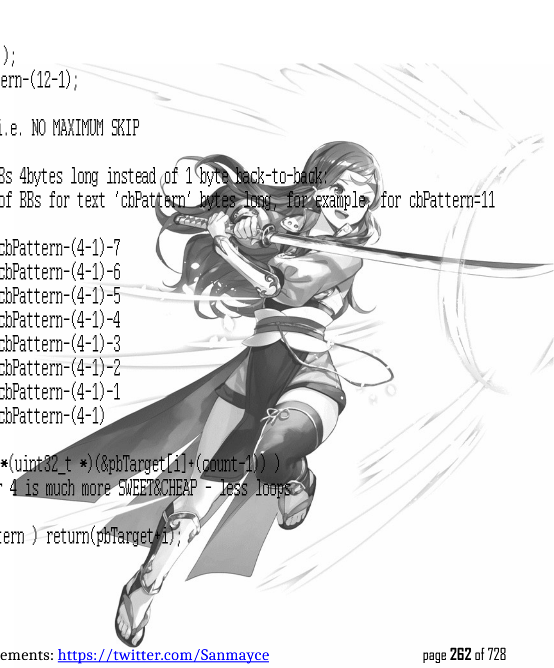
```
            Gulliver = 1;
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        // Order 4 ]
                }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        } // while (i <= cbTarget-cbPattern)
```

```
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_6 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
```

```
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
```



```
        for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        // Order 4 ]
                }
            i = i + Gulliver;
```



```
                        //GlobalI++; // Comment it, it is only for stats.
                } // while (i <= cbTarget-cbPattern)
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_7 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
```

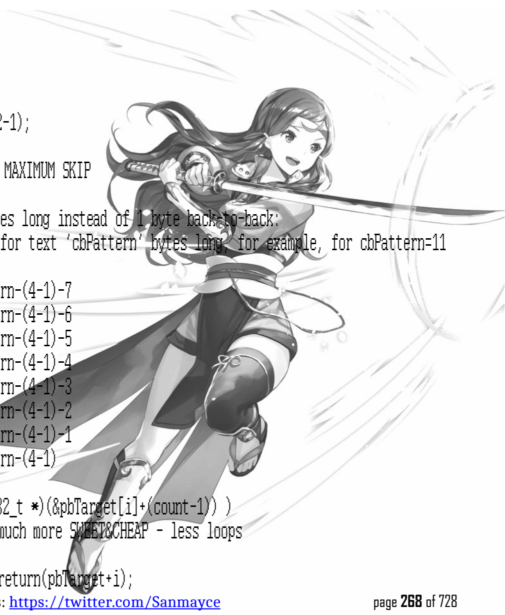
```
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
```

```
        //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
        for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
  0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
  'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //          if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
  ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
  } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

  } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
  cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
  'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        // Order 4 ]
```

```
                        }
                    i = i + Gulliver;
                    //GlobalI++; // Comment it, it is only for stats.
                } // while (i <= cbTarget-cbPattern)
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_8 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                    if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                        if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                    }
                    if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                        pbTarget++;
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                    }
                    pbTarget++;
                    if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                    count = cbPattern-1;
                    while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                        count--;
                    }
                    if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                        AdvanceHopperGrass++;
                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                            AdvanceHopperGrass++;
                            if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                        }
                    }
```

```
            }
            AdvanceHopperGrass++;
            pbTarget = pbTarget + AdvanceHopperGrass;
            if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
            //countSTATIC = cbPattern-2-2;
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
            i=0;
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
            //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
            for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
            while (i <= cbTarget-cbPattern) {
                Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            return(pbTarget+i);
                        }
                        //if ( count <= 0 ) {
                        //      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        //}
                        //else {
                        //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                        }
                    } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                } else Gulliver = cbPattern-(2-1);
                i = i + Gulliver;
                //GlobalI++; // Comment it, it is only for stats.
            }
            return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
            //countSTATIC = cbPattern-2-2;
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
            i=0;
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
            //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
            // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
            //"fast"
            //"aste"
            //"stes"
            //"test"
            //"est "
            //"st f"
```

```
        //"t fo"
        //" fox"
        //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
        for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //  if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )    Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
```



```
							}
							// Order 4 ]
					}
				i = i + Gulliver;
				//GlobalI++; // Comment it, it is only for stats.
			} // while (i <= cbTarget-cbPattern)
			return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
	} // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
		} //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
	} //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_9 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
	char * pbTargetMax = pbTarget + cbTarget;
	register uint32_t ulHashPattern;
	register uint32_t ulHashTarget;
	signed long count;
	//signed long countSTATIC;

	unsigned char SINGLET;
	uint32_t Quadruplet2nd;
	uint32_t Quadruplet3rd;
	uint32_t Quadruplet4th;

	uint32_t AdvanceHopperGrass;

	uint32_t a, i, j;
//Global is next line already:
	//unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
	uint32_t Gulliver; // or unsigned char or unsigned short

	//unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
	uint32_t hash32;
	uint32_t hash32B;
	uint32_t hash32C;

	if (cbPattern > cbTarget) return(NULL);

	if ( cbPattern<4 ) {

		pbTarget = pbTarget+cbPattern;
		ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
		if ( cbPattern==3 ) {
			for ( ;; ) {
				if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
					if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
				}
				if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
					pbTarget++;
					if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
				}
				pbTarget++;
				if (pbTarget > pbTargetMax) return(NULL);
			}
		} else {
		}
		for ( ;; ) {
			if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
			if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
			pbTarget++;
			if (pbTarget > pbTargetMax) return(NULL);
		}

	} else {
		if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

		pbTarget = pbTarget+cbPattern;
		ulHashPattern = *(uint32_t *)(pbPattern);
		SINGLET = ulHashPattern & 0xFF;
		Quadruplet2nd = SINGLET<<8;
		Quadruplet3rd = SINGLET<<16;
		Quadruplet4th = SINGLET<<24;
		for ( ;; ) {
			AdvanceHopperGrass = 0;
			ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
			if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
				count = cbPattern-1;
				while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
					if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
					count--;
				}
				if ( count == 0) return((pbTarget-cbPattern));
			} else { // The goal here: to avoid memory accesses by stressing the registers.
				if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
					AdvanceHopperGrass++;
					if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
						AdvanceHopperGrass++;
						if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
```

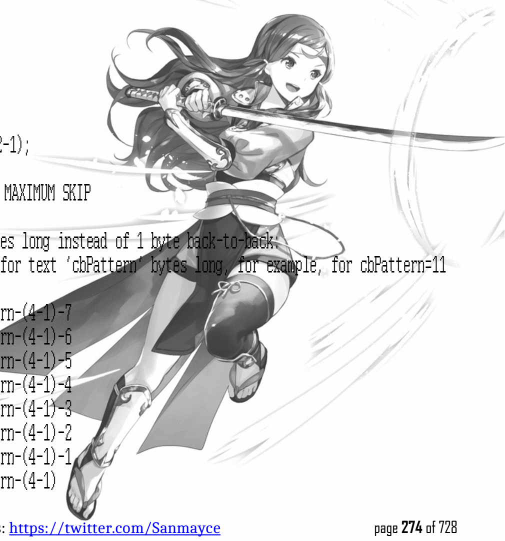
```
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
```

```
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
                for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                        // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                                if ( count <= 0 ) {
                                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                }
                                                //else {
                                                //              if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                //}
                                                // Order 4 ]
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
                for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
                // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
                for (j=0; j < cbPattern-12+1; j++) {
                        hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
                        hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
                        hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
                        hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                        hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                        hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                        bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
                }
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // Assume minimal jump as initial value.
                        // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
                        //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                                else {
                                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                        // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
```

```
                                                if ( count <= 0 ) {
                                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                }
                                                // Order 4 ]
                                }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                } // while (i <= cbTarget-cbPattern)
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_0 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
```

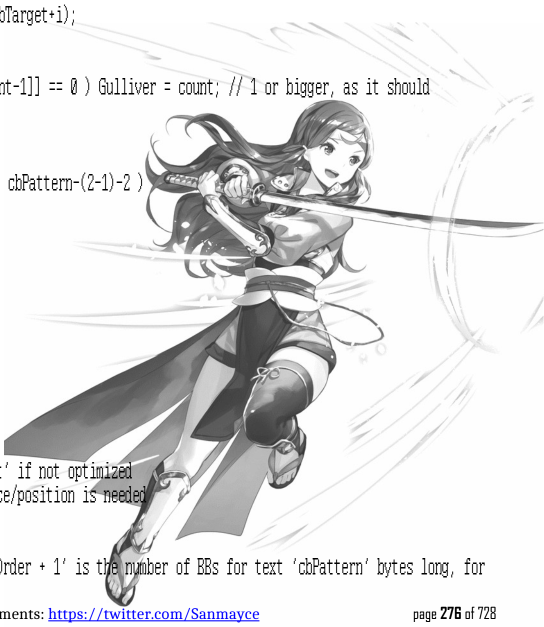
```
                        AdvanceHopperGrass++;
                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                    }
                }
            }
            AdvanceHopperGrass++;
            pbTarget = pbTarget + AdvanceHopperGrass;
            if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
            //countSTATIC = cbPattern-2-2;
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
            i=0;
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
            //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
            for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
            while (i <= cbTarget-cbPattern) {
                Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            return(pbTarget+i);
                        }
                        //if ( count <= 0 ) {
                        //  if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        //}
                        //else {
                        //  if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                        }
                    } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                } else Gulliver = cbPattern-(2-1);
                i = i + Gulliver;
                //GlobalI++; // Comment it, it is only for stats.
            }
            return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
            //countSTATIC = cbPattern-2-2;
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
            i=0;
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
            //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
            // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
            //"fast"
            //"aste"
```

```
		//"stes"
		//"test"
		//"est "
		//"st f"
		//"t fo"
		//" fox"
		//for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
		for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
		while (i <= cbTarget-cbPattern) {
			Gulliver = 1;
			if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
				if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = cbPattern-(2-1)-2-4; else {
						// Order 4 [
					// Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
					// Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
					//0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
					//1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
					//2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
					//3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
					//4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
					//5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
					//6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
					//7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
						count = cbPattern-4+1;
						while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
							count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
						if ( count <= 0 ) {
							if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
						}
						//else {
						//	if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
						//}
						// Order 4 ]
				}
			} else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
			i = i + Gulliver;
			//GlobalI++; // Comment it, it is only for stats.
		}
		return(NULL);
	} //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

	} else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
		//countSTATIC = cbPattern-2-2;
		ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
		//ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
		i=0;
		for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
		// cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
		for (j=0; j < cbPattern-12+1; j++) {
			hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
			hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
			hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
			hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
			hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
			hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
			bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
		}
		while (i <= cbTarget-cbPattern) {
			Gulliver = 1; // Assume minimal jump as initial value.
			// The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
			//GlobalHashSectionExecution++; // Comment it, it is only for stats.
				hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
				hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
				hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
				hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
				hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
				hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
				if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )	Gulliver = cbPattern-(12-1);
				else {
				//if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
						// Order 4 [
					// Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
					// Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
					//0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
					//1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
					//2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
					//3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
					//4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
					//5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
					//6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
					//7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
						count = cbPattern-4+1;
```



```
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                                if ( count <= 0 ) {
                                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                }
                                                // Order 4 ]
                                }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                } // while (i <= cbTarget-cbPattern)
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_a (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer Moore Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
```

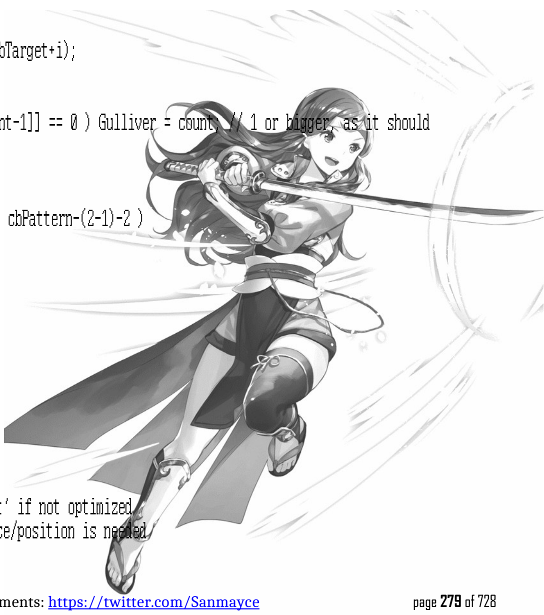
```
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }
                        AdvanceHopperGrass++;
                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax) return(NULL);
                }

                } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                        //countSTATIC = cbPattern-2-2;
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                        i=0;
                        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                        for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                        while (i <= cbTarget-cbPattern) {
                                Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                        // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                                if ( count <= 0 ) {
                                                        return(pbTarget+i);
                                                }
                                                //if ( count <= 0 ) {
                                                //              if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                //}
                                                //else {
                                                //              if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                //}
                                                // Order 4 ]
                                                }
                                        } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                                } else Gulliver = cbPattern-(2-1);
                                i = i + Gulliver;
                                //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                        //countSTATIC = cbPattern-2-2;
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                        i=0;
                        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
                        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
```



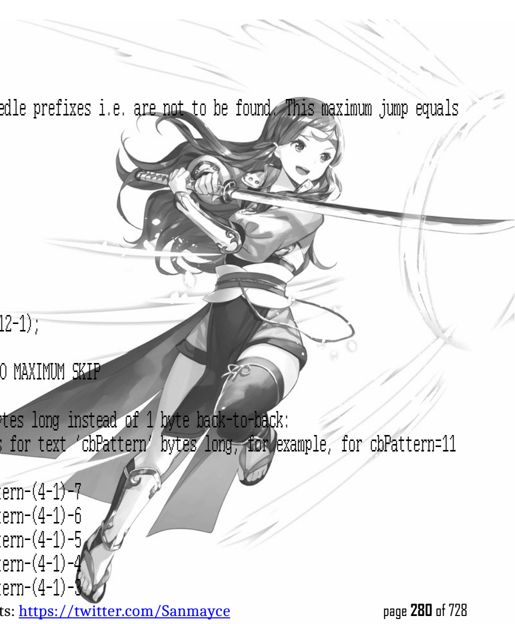
```
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
                for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] == 
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                        // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1; 
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                                if ( count <= 0 ) {
                                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                }
                                                //else {
                                                //      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) & 
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                //}
                                                // Order 4 ]
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
                for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
                // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
                for (j=0; j < cbPattern-12+1; j++) {
                        hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
                        hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
                        hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
                        hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                        hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                        hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                        bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
                }
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // Assume minimal jump as initial value.
                        // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals 
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
                        //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                                else {
                                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                        // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
```

```
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                                if ( count <= 0 ) {
                                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                }
                                                // Order 4 ]
                                }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                } // while (i <= cbTarget-cbPattern)
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_b (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
```

```
          } else { // The goal here: to avoid memory accesses by stressing the registers.
              if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                  AdvanceHopperGrass++;
                  if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                      AdvanceHopperGrass++;
                      if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                  }
              }
          }
          AdvanceHopperGrass++;
          pbTarget = pbTarget + AdvanceHopperGrass;
          if (pbTarget > pbTargetMax) return(NULL);
      }

      } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
          //countSTATIC = cbPattern-2-2;
          ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
          //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
          i=0;
          //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
          //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
          //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
          for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
          while (i <= cbTarget-cbPattern) {
              Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
              if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                  if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                      // Order 4 [
                  // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                  // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                  //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                  //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                  //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                  //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                  //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                  //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                  //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                  //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                      count = cbPattern-4+1;
                      //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                      while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                          count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                      if ( count <= 0 ) {
                          return(pbTarget+i);
                      }
                      //if ( count <= 0 ) {
                      //          if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                      //}
                      //else {
                      //          if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                      //}
                      // Order 4 ]
                      }
                  } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
              } else Gulliver = cbPattern-(2-1);
              i = i + Gulliver;
              //GlobalI++; // Comment it, it is only for stats.
          }
          return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
          //countSTATIC = cbPattern-2-2;
          ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
          //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
          i=0;
          //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
          //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
          //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
```

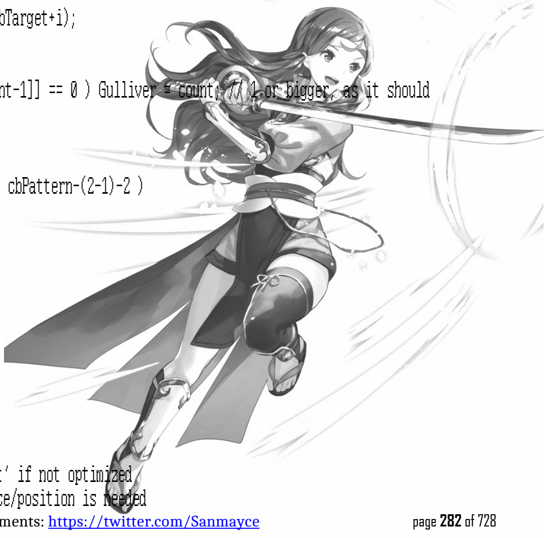


```
		// In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
	example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
		//"fast"
		//"aste"
		//"stes"
		//"test"
		//"est "
		//"st f"
		//"t fo"
		//" fox"
		//for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
		for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
		while (i <= cbTarget-cbPattern) {
			Gulliver = 1;
			if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
				if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
	0 ) Gulliver = cbPattern-(2-1)-2-4; else {
						// Order 4 [
					// Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
					// Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
	'fastest fox' and Order=4 we have BBs = 11-4+1=8:
					//0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
					//1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
					//2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
					//3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
					//4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
					//5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
					//6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
					//7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
						count = cbPattern-4+1;
						while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
							count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
						if ( count <= 0 ) {
							if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
						}
						//else {
						//	if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
	( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
						//}
						// Order 4 ]
				}
			} else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
			i = i + Gulliver;
			//GlobalI++; // Comment it, it is only for stats.
		}
		return(NULL);
	} //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

	} else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
		//countSTATIC = cbPattern-2-2;
		ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
		//ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
		i=0;
		for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
		// cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
		for (j=0; j < cbPattern-12+1; j++) {
			hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
			hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
			hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
			hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
			hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
			hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
			bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
		}
		while (i <= cbTarget-cbPattern) {
			Gulliver = 1; // Assume minimal jump as initial value.
			// The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
	cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
			//GlobalHashSectionExecution++; // Comment it, it is only for stats.
				hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
				hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
				hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
				hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
				hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
				hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
				if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
				else {
				//if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
						// Order 4 [
					// Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
					// Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
	'fastest fox' and Order=4 we have BBs = 11-4+1=8:
					//0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
					//1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
					//2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
					//3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
					//4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
```



```
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                                if ( count <= 0 ) {
                                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                }
                                                // Order 4 ]
                                }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                } // while (i <= cbTarget-cbPattern)
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_c (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, j, i;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
```

```
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //              if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //              if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
```



```
//Global is next line already:
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
        //"st f"
        //"t fo"
        //" fox"
        //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
        for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //  if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
```

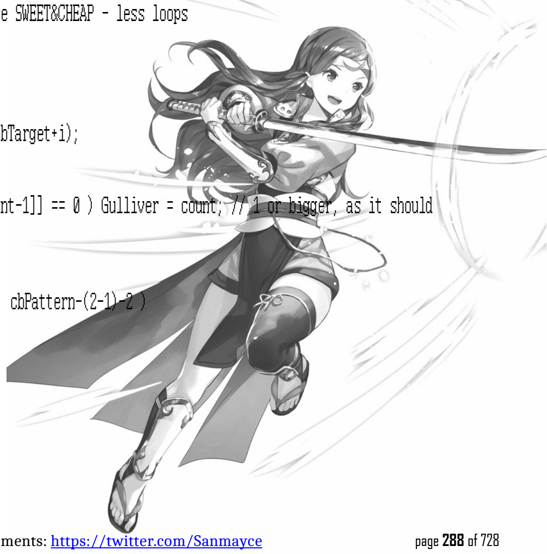
```
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                                if ( count <= 0 ) {
                                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                }
                                                // Order 4 ]
                                }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                } // while (i <= cbTarget-cbPattern)
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_d (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
```

```
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //              if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //              if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
                i=0;
```

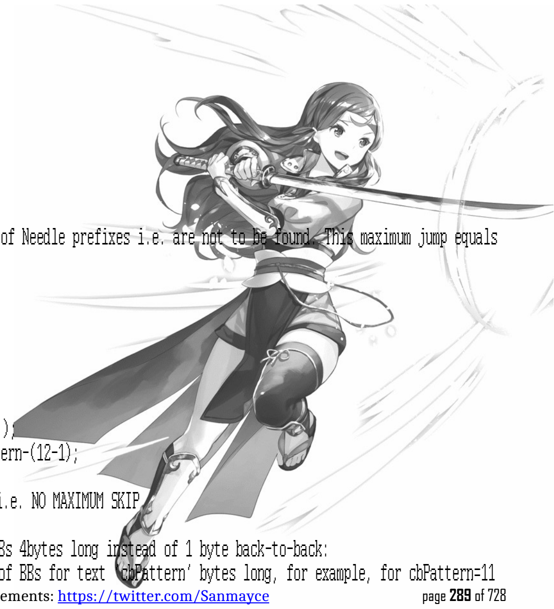
```
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
        //"st f"
        //"t fo"
        //" fox"
        //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
        for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //    if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
```

```
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                                if ( count <= 0 ) {
                                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                                }
                                                // Order 4 ]
                                }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                } // while (i <= cbTarget-cbPattern)
                return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_e (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
```

```
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
```

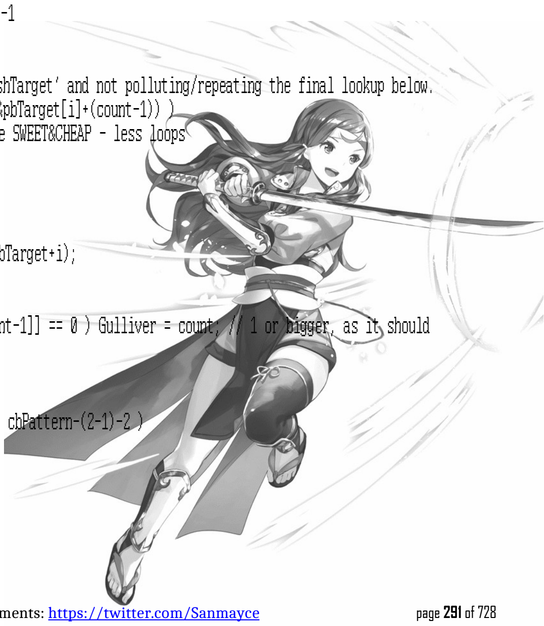


```
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
        //"st f"
        //"t fo"
        //" fox"
        //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
        for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )    Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
```



```
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                            count = cbPattern-4+1;
                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                            if ( count <= 0 ) {
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                            }
                            // Order 4 ]
                    }
                i = i + Gulliver;
                //GlobalI++; // Comment it, it is only for stats.
            } // while (i <= cbTarget-cbPattern)
            return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
    } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
        } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    } //if ( cbPattern<4 )
}
char * Railgun_Sekireigan_Wolfram_f (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register uint32_t ulHashPattern;
    register uint32_t ulHashTarget;
    signed long count;
    //signed long countSTATIC;

    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;

    uint32_t AdvanceHopperGrass;

    uint32_t a, i, j;
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; // BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    uint32_t Gulliver; // or unsigned char or unsigned short

    //unsigned char bm_Hasherezade_HASH[1<<(HashTableSizeSekireiTchittoGritto-3)];
    uint32_t hash32;
    uint32_t hash32B;
    uint32_t hash32C;

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
            for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                    if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                }
                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                    pbTarget++;
                    if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                }
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
            }
        } else {
        }
        for ( ;; ) {
            if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
            if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
            pbTarget++;
            if (pbTarget > pbTargetMax) return(NULL);
        }

    } else {
        if (cbTarget<HaystackThresholdSekireiTchittoGritto) { // This value is arbitrary (don't know how exactly), it ensures (at least must) better performance than 'Boyer_Moore_Horspool'.

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;
        for ( ;; ) {
            AdvanceHopperGrass = 0;
```

```
                ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
                if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                        count = cbPattern-1;
                        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                count--;
                        }
                        if ( count == 0) return((pbTarget-cbPattern));
                } else { // The goal here: to avoid memory accesses by stressing the registers.
                        if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                AdvanceHopperGrass++;
                                if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                }
                        }
                }
                AdvanceHopperGrass++;
                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax) return(NULL);
        }

        } else { //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
    if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto ) {

    // BMH order 2:
    if ( cbPattern<=NeedleThreshold2vs4TchittoGritto ) {
                //countSTATIC = cbPattern-2-2;
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //ulHashTarget = *(uint32_t *)(pbPattern+cbPattern-4); // Last four bytes
                i=0;
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//Possible commenting of next line:
                for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=1;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
    // Few thoughts regarding an excellent Skip Performance etude:
    // Something "still" considered crazy: using BITwise order 3, not pseudo order 3, though!
    // 2^24 = 16MB BYTEwise or 2^(24-3) = 2MB BITwise, when searching big haystacks e.g. Wikipedia 42GB with Kazahana then this 2MB lookup table seems not so atrocious.
    // The code is like:
    // ulHashTarget = *(uint32_t *)&pbTarget[i+cbPattern-4]; // One memory access instead of 2
    // if ( bm_Horspool_Order3[ulHashTarget>>8] == 0 ) Gulliver = cbPattern-(3-1); else
    // if ( bm_Horspool_Order3[ulHashTarget&0xFFFFFF] == 0 ) Gulliver = cbPattern-(3-1)-1; else
    // {
    // ...
    // }
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-1]] != 2 )
    Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                        // Order 4 [
                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox'
    and Order=4 we have BBs = 11-4+1=8:
                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                        count = cbPattern-4+1;
                                        //count = count-4; // Double-beauty here of already being checked 'ulHashTarget' and not polluting/repeating the final lookup below.
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                                        if ( count <= 0 ) {
                                                return(pbTarget+i);
                                        }
                                        //if ( count <= 0 ) {
                                        //              if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                                        //}
                                        //else {
                                        //              if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                        //}
                                        // Order 4 ]
                                        }
                                } // Means AT LEAST one of the BBs is 0, enforce lower skip: MIN( cbPattern-(2-1)-1 , cbPattern-(2-1)-2 )
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
    // BMH order 4, needle should be >=8:
    } else { //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )
```



```
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
//Global is next line already:
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
        //"st f"
        //"t fo"
        //" fox"
        //for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+j+0) + *(unsigned short *)(pbPattern+j+2) ) & ( (1<<16)-1 )]=1;
//Possible commenting of next line:
        for (j=0; j < cbPattern-4+1; j++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+j+0)>>16)+(*(uint32_t *)(pbPattern+j+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) {
                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        //else {
                        //  if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) ) &
    ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                        //}
                        // Order 4 ]
                }
            } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
    } //if ( cbPattern<=NeedleThreshold2vs4TchittoGritto )

    } else { // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() [
        //countSTATIC = cbPattern-2-2;
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes
        i=0;
        for (a=0; a < 1<<(HashTableSizeSekireiTchittoGritto-3); a++) {bm_Hasherezade_HASH[a]= 0;} // to-do: 'memset' if not optimized
        // cbPattern - Order + 1 i.e. number of BBs for 11 'fastest fox' 11-8+1=4: 'fastest ', 'astest f', 'stest fo', 'test fox'
        for (j=0; j < cbPattern-12+1; j++) {
            hash32 = (2166136261UL ^ *(uint32_t *)(pbPattern+j+0)) * 709607;
            hash32B = (2166136261UL ^ *(uint32_t *)(pbPattern+j+4)) * 709607;
            hash32C = (2166136261UL ^ *(uint32_t *)(pbPattern+j+8)) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
            hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
            hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
            bm_Hasherezade_HASH[hash32>>3]= bm_Hasherezade_HASH[hash32>>3] | (1<<(hash32&0x7));
        }
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // Assume minimal jump as initial value.
            // The goal: to jump when the rightmost 8bytes (Order 8 Horspool) of window do not look like any of Needle prefixes i.e. are not to be found. This maximum jump equals
    cbPattern-(Order-1) or 11-(8-1)=4 for 'fastest fox' - a small one but for Needle 31 bytes the jump equals 31-(8-1)=24
            //GlobalHashSectionExecution++; // Comment it, it is only for stats.
                hash32 = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+0)) * 709607;
                hash32B = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+4)) * 709607;
                hash32C = (2166136261UL ^ *(uint32_t *)(pbTarget+i+cbPattern-12+8)) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32C,5) ) * 709607;
                hash32 = (hash32 ^ _rotl_KAZE(hash32B,5) ) * 709607;
                hash32 = ( hash32 ^ (hash32 >> 16) ) & ( (1<<(HashTableSizeSekireiTchittoGritto))-1 );
                if ( (bm_Hasherezade_HASH[hash32>>3] & (1<<(hash32&0x7))) ==0 )   Gulliver = cbPattern-(12-1);
                else {
                //if ( Gulliver == 1 ) { // Means the Building-Block order 8/12 is found somewhere i.e. NO MAXIMUM SKIP
                        // Order 4 [
```

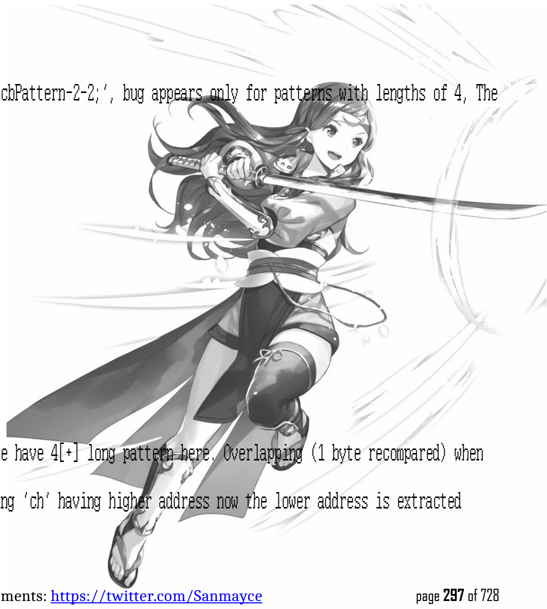
```
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
  'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4; // - order, of course order 4 is much more SWEET&CHEAP - less loops
                        if ( count <= 0 ) {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern ) return(pbTarget+i);
                        }
                        // Order 4 ]
                }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        } // while (i <= cbTarget-cbPattern)
        return(NULL);
// MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() MEMMEM() ]
  } // if ( cbPattern<=NeedleThresholdBIGSekireiTchittoGritto )
    } //if (cbTarget<HaystackThresholdSekireiTchittoGritto)
  } //if ( cbPattern<4 )
}

*/
// Commented 2019-May-17 ]]]]]]]]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_1 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
```

```
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/



        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//       countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
```

```
          count = cbPattern-1;
          while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
               if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                count--;
          }
          if ( count == 0) return((pbTarget-cbPattern));
         } else { // The goal here: to avoid memory accesses by stressing the registers.
      if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
          AdvanceHopperGrass++;
          if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
               AdvanceHopperGrass++;
               if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
          }
      }
         }

        AdvanceHopperGrass++;

        pbTarget = pbTarget + AdvanceHopperGrass;
         if (pbTarget > pbTargetMax)
             return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_2 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
```

```
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
```

```
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
```



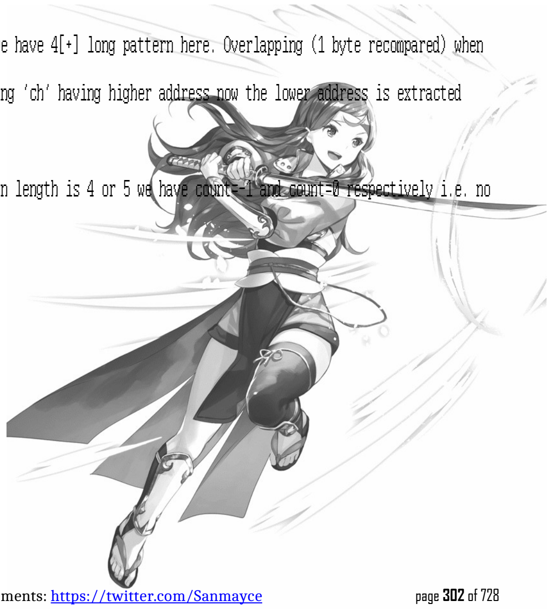
```
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
        need of comparing in-between chars.
              count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_3 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
        strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock
```

```
/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise 368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;
```

```
       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_4 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
```

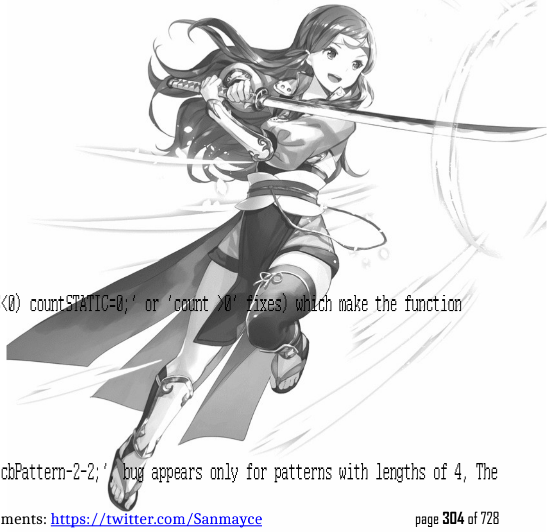
```
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
```

```
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//               if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                 count--;
//          }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
               if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
        count = countSTATIC;
        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
              count--;
        }
        if ( count == 0) return(pbTarget+i);
           }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
```



```
char * Railgun_Quadruplet_7_5 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/
```

```
        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
```



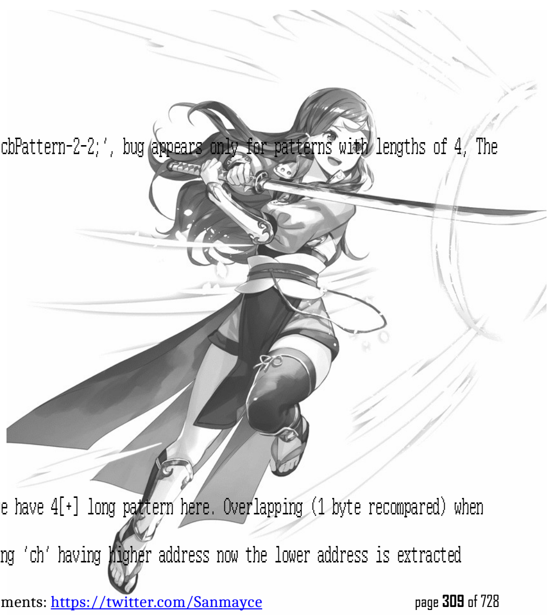
```
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_6 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
```

```
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//           count = countSTATIC;
         count = cbPattern-2;
//           while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/



        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//          countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//           count = countSTATIC;
//           while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//               if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
```

```
//              count--;
//         }
          count = cbPattern-1;
          while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
               if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                count--;
          }
          if ( count == 0) return((pbTarget-cbPattern));
         } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
          AdvanceHopperGrass++;
          if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
               AdvanceHopperGrass++;
               if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
          }
    }
        }

        AdvanceHopperGrass++;

        pbTarget = pbTarget + AdvanceHopperGrass;
         if (pbTarget > pbTargetMax)
             return(NULL);
    }
} else { //if (cbTarget<961)
         //countSTATIC = cbPattern-2; //r.6+
         //countSTATIC = cbPattern-2-3;
         //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
        unfittable in code cache lines?!
         //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
         countSTATIC = cbPattern-2-2; // r.7
         ulHashPattern = *(unsigned long *)(pbPattern);

         //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
        setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
        length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
        first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
          count = countSTATIC;
          while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
        need of comparing in-between chars.
                count--;
          }
          if ( count == 0) return(pbTarget+i);
             }
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_7 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
```

```
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//          countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//           count = countSTATIC;
         count = cbPattern-2;
//           while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
```

```
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
               if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
```



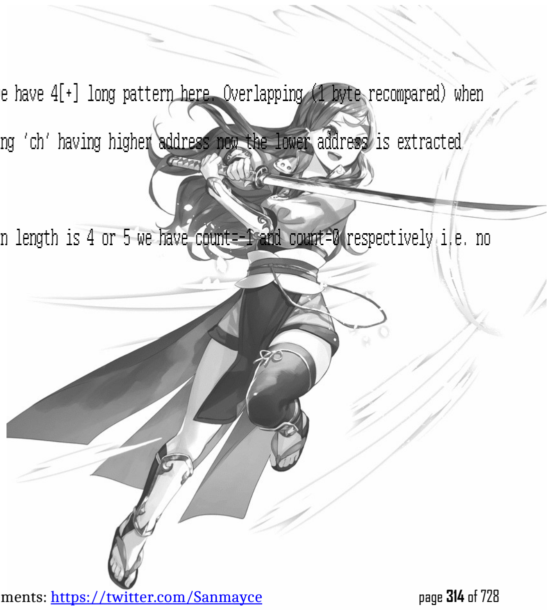
```
            {
        count = countSTATIC;
        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
      need of comparing in-between chars.
              count--;
        }
        if ( count == 0) return(pbTarget+i);
           }
      i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_8 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
```

```
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/



        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//              count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }
```

```
        AdvanceHopperGrass++;

        pbTarget = pbTarget + AdvanceHopperGrass;
         if (pbTarget > pbTargetMax)
             return(NULL);
    }
} else { //if (cbTarget<961)
         //countSTATIC = cbPattern-2; //r.6+
         //countSTATIC = cbPattern-2-3;
         //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
        unfittable in code cache lines?!
         //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
         countSTATIC = cbPattern-2-2; // r.7
         ulHashPattern = *(unsigned long *)(pbPattern);

         //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
        setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_9 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed
```

```
    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;



        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
```

```
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//Global1++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]
```



```
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_0 (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//          count = countSTATIC;
         count = cbPattern-2;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/
```

```
        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
```

```
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_a (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
```

```
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//           count = countSTATIC;
         count = cbPattern-2;
//           while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//           count = countSTATIC;
```

```
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//         }
          count = cbPattern-1;
          while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
          }
          if ( count == 0) return((pbTarget-cbPattern));
         } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
        }

        AdvanceHopperGrass++;

        pbTarget = pbTarget + AdvanceHopperGrass;
         if (pbTarget > pbTargetMax)
             return(NULL);
    }
} else { //if (cbTarget<961)
         //countSTATIC = cbPattern-2; //r.6+
         //countSTATIC = cbPattern-2-3;
         //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
        unfittable in code cache lines?!
         //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
         countSTATIC = cbPattern-2-2; // r.7
         ulHashPattern = *(unsigned long *)(pbPattern);

         //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
        setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
        length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
        first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
              {
          count = countSTATIC;
          while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
        need of comparing in-between chars.
               count--;
          }
          if ( count == 0) return(pbTarget+i);
             }
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_b (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
```

```
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
```

```
          }
         //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
        368KB/clock.
          while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
                countRemainder--;
          }
          //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
          if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
          //}
         }
*/

         pbTarget++;
         if (pbTarget > pbTargetMax)
             return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
         pbTarget = pbTarget+cbPattern;
         ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
        AdvanceHopperGrass = 0;
        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

         if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//              if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//          }
          count = cbPattern-1;
          while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                 count--;
          }
          if ( count == 0) return((pbTarget-cbPattern));
         } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
          AdvanceHopperGrass++;
          if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                AdvanceHopperGrass++;
                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
          }
    }
        }

        AdvanceHopperGrass++;

        pbTarget = pbTarget + AdvanceHopperGrass;
         if (pbTarget > pbTargetMax)
             return(NULL);
    }
} else { //if (cbTarget<961)
         //countSTATIC = cbPattern-2; //r.6+
         //countSTATIC = cbPattern-2-3;
         //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
        unfittable in code cache lines?!
         //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
         countSTATIC = cbPattern-2-2; // r.7
         ulHashPattern = *(unsigned long *)(pbPattern);

         //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
        setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
        length=4, grmbl.
```

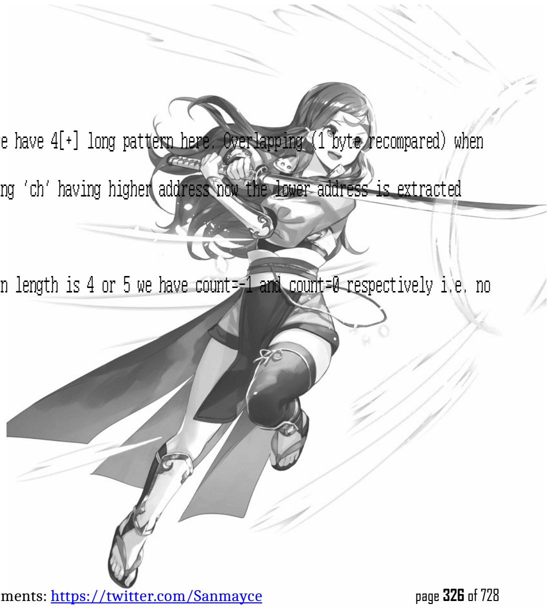
```
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
            {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
              count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_c (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/
```

```
        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//               if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                 count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
               if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
```

```
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_d (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
```

```
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
```

```
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
```



```
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_e (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//         count = countSTATIC;
         count = cbPattern-2;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
```

```
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//          }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);
```

```
        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
             {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm [
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Quadruplet_7_f (char * pbTarget,
     char * pbPattern,
     unsigned long cbTarget,
     unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long  ulHashPattern;
    unsigned long ulHashTarget;
    //unsigned long count; //r.6+
    signed long count;
    //unsigned long countSTATIC; //r.6+
    signed long countSTATIC;
//  unsigned long countRemainder;

/*
    const unsigned char SINGLET = *(char *)(pbPattern);
    const unsigned long Quadruplet2nd = SINGLET<<8;
    const unsigned long Quadruplet3rd = SINGLET<<16;
    const unsigned long Quadruplet4th = SINGLET<<24;
*/
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
//Below array is already global:
    int a, j;
    //int a, j, bm_bc[ASIZE]; //BMH needed
    unsigned char ch; //BMH needed
    unsigned long chchchch; //BMH needed
//    unsigned char lastch, firstch; //BMH needed

    if (cbPattern > cbTarget)
        return(NULL);

// Doesn't work when cbPattern = 1
// The next IF-fragment works very well with cbPattern>1, OBVIOUSLY IT MUST BE UNROLLED(but crippled with less functionality) SINCE either cbPattern=2 or cbPattern=3!
if ( cbPattern<4) { // This IF makes me unhappy: it slows down from 390KB/clock to 367KB/clock for 'fast' pattern. This fragment(for 2..3 pattern lengths) is needed because I need a function different than
       strchr but sticking to strstr i.e. lengths above 1 are to be handled.
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
//        countSTATIC = cbPattern-2;

if ( cbPattern==3) {
    for ( ;; )
    {
```

```
        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
         if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else {
}
    for ( ;; )
    {
        // The line below gives for 'cbPattern'>=1:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/543
        // Karp_Rabin_Kaze_4_OCTETS performance: 372KB/clock
/*
        if ( (ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1)) && !memcmp(pbPattern, pbTarget-cbPattern, (unsigned int)cbPattern) )
            return((long)(pbTarget-cbPattern));
*/

        // The fragment below gives for 'cbPattern'>=2:
        // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/546
        // Karp_Rabin_Kaze_4_OCTETS performance: 370KB/clock

/*
//For 2 and 3 [
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
//           count = countSTATIC;
         count = cbPattern-2;
//           while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
         while ( count && *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) { // Crippling i.e. only 2 and 3 chars are allowed!
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        }
        if ( (char)(ulHashPattern>>8) != *(pbTarget-cbPattern+1) ) pbTarget++;
//For 2 and 3 ]
*/


        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
            return((pbTarget-2));
        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;


        // The fragment below gives for 'cbPattern'>=2:
       // Karp_Rabin_Kaze_4_OCTETS_hits/Karp_Rabin_Kaze_4_OCTETS_clocks: 4/554
       // Karp_Rabin_Kaze_4_OCTETS performance: 364KB/clock
/*
        if ( ulHashPattern == ( (*(char *)(pbTarget-cbPattern))<<8 ) + *(pbTarget-1) ) {
         count = countSTATIC>>2;
         countRemainder = countSTATIC % 4;

         while ( count && *(unsigned long *)(pbPattern+1+((count-1)<<2)) == *(unsigned long *)(pbTarget-cbPattern+1+((count-1)<<2)) ) {
               count--;
         }
        //if (count == 0) {  // Disastrous degradation only from this line(317KB/clock when 1+2x4+2+1 bytes pattern: 'skillessness'; 312KB/clock when 1+1x4+2+1 bytes pattern: 'underdog'), otherwise
       368KB/clock.
         while ( countRemainder && *(char *)(pbPattern+1+(countSTATIC-countRemainder)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-countRemainder)) ) {
               countRemainder--;
         }
         //if ( countRemainder == 0) return((long)(pbTarget-cbPattern));
         if ( count+countRemainder == 0) return((long)(pbTarget-cbPattern));
         //}
        }
*/

        pbTarget++;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if ( cbPattern<4)
if (cbTarget<961) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(unsigned long *)(pbPattern);
//          countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);
```

```
        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//          }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else { //if (cbTarget<961)
        //countSTATIC = cbPattern-2; //r.6+
        //countSTATIC = cbPattern-2-3;
        //countSTATIC = cbPattern-2-2; // r.6+++ I suppose that the awful degradation comes from 2bytes more (from either 'if (countSTATIC<0) countSTATIC=0;' or 'count >0' fixes) which make the function
       unfittable in code cache lines?!
        //countSTATIC = cbPattern-2-3; // r.7- At last no recompared bytes in-between chars
        countSTATIC = cbPattern-2-2; // r.7
        ulHashPattern = *(unsigned long *)(pbPattern);

        //chPTR=(unsigned char *)&chchchch+3;
// Next line fixes the BUG from r.6++: but with awful speed degradation! So the bug is fixed in the definitions by setting 'countSTATIC = cbPattern-2-2;', bug appears only for patterns with lengths of 4, The
       setback is one unnecessary comparison for 5bytes patterns, stupidly such setback exists (from before) for 4bytes as well.
//if (countSTATIC<0) countSTATIC=0;
    /* Preprocessing */
//Below 2 lines are global already:
    //for (a=0; a < ASIZE; a++) bm_bc[a]=cbPattern;
    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;

    /* Searching */
    //lastch=pbPattern[cbPattern-1];
    //firstch=pbPattern[0];
    i=0;
    while (i <= cbTarget-cbPattern) {
       //ch=pbTarget[i+cbPattern-1];
       //ch=pbTarget[i];
          //if ( pbTarget[i] == pbPattern[0] && *(unsigned long *)&pbTarget[i+cbPattern-1-3] == ulHashPattern) // No problema here since we have 4[+] long pattern here. Overlapping (1 byte recompared) when
       length=4, grmbl.
          if ( *(unsigned long *)&pbTarget[i] == ulHashPattern ) // The lesson I learned from r.7- now applied in r.7: instead of extracting 'ch' having higher address now the lower address is extracted
       first in order (hopefully, the test confirms it) the next 32bytes (including 'ch') to be cached i.e. to comparison part is faster.
            {
         count = countSTATIC;
         while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no
       need of comparing in-between chars.
               count--;
         }
         if ( count == 0) return(pbTarget+i);
            }
       i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
//GlobalI++;
    }
    return(NULL);
} //if (cbTarget<961)
} //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Horspool algorithm ]

// Commented 2019-May-17 [[[[[[[[
/*

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_1 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
       char * pbTargetMax = pbTarget + cbTarget;
       register unsigned long ulHashPattern;
       register unsigned long ulHashTarget;
       signed long count;
       signed long countSTATIC;

       unsigned char SINGLET;
```

```
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                     if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                     } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;
```

```
                           // Elsiane r.2  [
//Global is next line already:
                           //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                           // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                           // fast 4 0-1-2 fa as st
//Global is next line already:
                           //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                           // Elsiane r.2 ]

                           ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                           //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                           i=0;
       if ( cbPattern>10) { // r.2
                           while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                      if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                                   count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                                                   while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                                                count--;
                                                                   if ( count == 0) return(pbTarget+i);
                                                      }
                                                      //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
         i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
         i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                                      i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                                        else
                                                      i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                                        //GlobalI++;
                           }
       } else { // r.2
                           while (i <= cbTarget-cbPattern-1) {
                                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                      count = countSTATIC;
                                                      while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
       have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                                                   count--;
                                                      }
                                                      if ( count == 0) return(pbTarget+i);
                                        }
                                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                                        //GlobalI++;
                           }
       } // r.2

                           if (i == cbTarget-cbPattern) {
                                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                      count = countSTATIC;
                                                      while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                                   count--;
                                                      if ( count == 0) return(pbTarget+i);
                                        }
                                        //GlobalI++;
                           }

                           return(NULL);
              } //if (cbTarget<961)
       } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
```

```
//
char * Railgun_Quadruplet_7Gulliver_2 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
```

```
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//      longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
```

```
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_3 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
```

```
                                                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                                }
                                        }
                                }

                                AdvanceHopperGrass++;

                                pbTarget = pbTarget + AdvanceHopperGrass;
                                if (pbTarget > pbTargetMax)
                                        return(NULL);
                        }

                } else { //if (cbTarget<961)
                        countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                        //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                        //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                        //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                        // Elsiane r.2  [
//Global is next line already:
                        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                        // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                        // fast 4 0-1-2 fa as st
//Global is next line already:
                        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                        // Elsiane r.2 ]

                        ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                        i=0;
    if ( cbPattern>10) { // r.2
                        while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                                Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                                if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                        count--;
                                                if ( count == 0) return(pbTarget+i);
                                        }
                                        //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                                } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                        i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                                else
                                        i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                                //GlobalI++;
                        }
    } else { // r.2
                        while (i <= cbTarget-cbPattern-1) {
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC;
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                                count--;
                                        }
                                        if ( count == 0) return(pbTarget+i);
                                }
                                i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                                //GlobalI++;
                        }
```

```
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_4 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
```

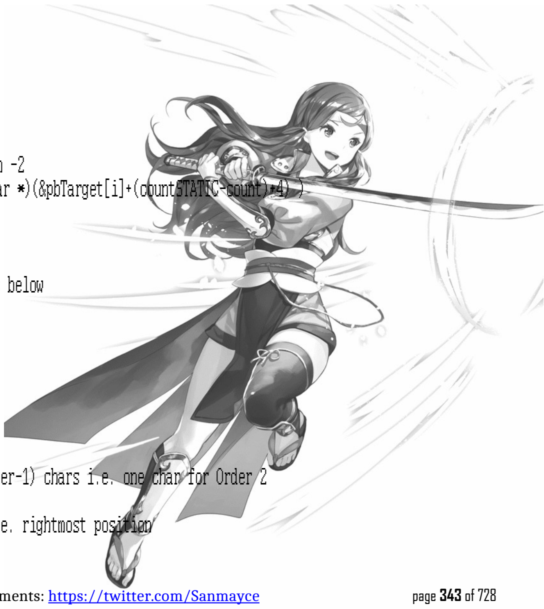
```
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                     } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
     i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
     i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
```

```
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_5 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
```

```
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
```

```
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//       longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_6 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
```

```
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
      i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
      i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
```



```
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 ’a’ vs ’t’
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 ’fa’ vs ’st’ i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 ’a’ vs ’e’
//       longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 ’fa’ vs ’ce’ i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case ’if (cbPattern==1)’ was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_7 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, ’char’ limits patterns to 255, if ’long’ then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
```

```
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                     if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                     } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
```

```
     i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
     i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//      longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_8 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
```

```
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
            for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                    if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                }
                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                    return(NULL);
            }
        } else {
        }
        for ( ;; ) {
            if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                return((pbTarget-2));
            if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
            pbTarget++;
            if (pbTarget > pbTargetMax)
                return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

            pbTarget = pbTarget+cbPattern;
            ulHashPattern = *(unsigned long *)(pbPattern);

            SINGLET = ulHashPattern & 0xFF;
            Quadruplet2nd = SINGLET<<8;
            Quadruplet3rd = SINGLET<<16;
            Quadruplet4th = SINGLET<<24;

            for ( ;; ) {
                AdvanceHopperGrass = 0;
                ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

              if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                    count = cbPattern-1;
                    while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                        count--;
                    }
                    if ( count == 0) return((pbTarget-cbPattern));
              } else { // The goal here: to avoid memory accesses by stressing the registers.
                    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                        AdvanceHopperGrass++;
                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                            AdvanceHopperGrass++;
                            if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                        }
                    }
                }

                AdvanceHopperGrass++;

                pbTarget = pbTarget + AdvanceHopperGrass;
                if (pbTarget > pbTargetMax)
                    return(NULL);
            }

        } else { //if (cbTarget<961)
            countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
            //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
            //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
            //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

            // Elsiane r.2  [
//Global is next line already:
            //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

            // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
            // fast 4 0-1-2 fa as st
//Global is next line already:
            //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

            // Elsiane r.2 ]

            ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
            //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

            i=0;
    if ( cbPattern>10) { // r.2
            while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
```

```
                                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                                count--;
                                                        if ( count == 0) return(pbTarget+i);
                                                }
                                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                                        else
                                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                                        //GlobalI++;
                                }
        } else { // r.2
                                while (i <= cbTarget-cbPattern-1) {
                                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                count = countSTATIC;
                                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
        have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                                        count--;
                                                }
                                                if ( count == 0) return(pbTarget+i);
                                        }
                                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                                        //GlobalI++;
                                }
        } // r.2

                                if (i == cbTarget-cbPattern) {
                                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                count = countSTATIC;
                                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                        count--;
                                                if ( count == 0) return(pbTarget+i);
                                        }
                                        //GlobalI++;
                                }

                                return(NULL);
                } //if (cbTarget<961)
        } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_9 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        register unsigned long ulHashPattern;
        register unsigned long ulHashTarget;
        signed long count;
        signed long countSTATIC;

        unsigned char SINGLET;
        unsigned long Quadruplet2nd;
        unsigned long Quadruplet3rd;
        unsigned long Quadruplet4th;

        unsigned long  AdvanceHopperGrass;

        long i; //BMH needed
        int a, j;
//Global are next 2 lines already:
```

```
        //unsigned int bm_bc[256]; //BMH needed
        //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
        //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
        unsigned long Gulliver; // or unsigned char or unsigned short

        if (cbPattern > cbTarget)
            return(NULL);

        if ( cbPattern<4) {
            pbTarget = pbTarget+cbPattern;
            ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

            if ( cbPattern==3) {
                    for ( ;; ) {
                            if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                    if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                            }
                            if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                            pbTarget++;
                            if (pbTarget > pbTargetMax)
                                    return(NULL);
                    }
            } else {
            }
            for ( ;; ) {
                    if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                            return((pbTarget-2));
                    if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                    pbTarget++;
                    if (pbTarget > pbTargetMax)
                            return(NULL);
            }
        } else { //if ( cbPattern<4)
            if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                    pbTarget = pbTarget+cbPattern;
                    ulHashPattern = *(unsigned long *)(pbPattern);

                    SINGLET = ulHashPattern & 0xFF;
                    Quadruplet2nd = SINGLET<<8;
                    Quadruplet3rd = SINGLET<<16;
                    Quadruplet4th = SINGLET<<24;

                    for ( ;; ) {
                            AdvanceHopperGrass = 0;
                            ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                    count = cbPattern-1;
                                    while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                            if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                            count--;
                                    }
                                    if ( count == 0) return((pbTarget-cbPattern));
                        } else { // The goal here: to avoid memory accesses by stressing the registers.
                                    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                            AdvanceHopperGrass++;
                                            if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                    AdvanceHopperGrass++;
                                                    if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                            }
                                    }
                            }

                            AdvanceHopperGrass++;

                            pbTarget = pbTarget + AdvanceHopperGrass;
                            if (pbTarget > pbTargetMax)
                                    return(NULL);
                    }

            } else { //if (cbTarget<961)
                    countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                    //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                    //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                    //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                    // Elsiane r.2  [
//Global is next line already:
                    //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                    // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                    // fast 4 0-1-2 fa as st
//Global is next line already:
                    //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed
```

```
                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//       longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_0 (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;
```

```
    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, ‘char’ limits patterns to 255, if ‘long’ then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don’t know how exactly), it ensures(at least must) better performance than ‘Boyer_Moore_Horspool’.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but ‘AdvanceHopperGrass’ must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;
```

```
                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // ‘memset’ if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to ‘SunHorse’: lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
     i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
     i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 ‘a’ vs ‘t’
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 ‘fa’ vs ‘st’ i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 ‘a’ vs ‘e’
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 ‘fa’ vs ‘ce’ i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
```

```
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_a (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
```

```
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
```

```
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_b (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
```

```
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
     i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
     i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
```

```
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_c (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                     if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
```

```
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                     } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
```

```
                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                    count = countSTATIC;
                    while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
  have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                        count--;
                    }
                    if ( count == 0) return(pbTarget+i);
                }
                i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                //GlobalI++;
            }
  } // r.2

            if (i == cbTarget-cbPattern) {
                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                    count = countSTATIC;
                    while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                        count--;
                    if ( count == 0) return(pbTarget+i);
                }
                //GlobalI++;
            }

            return(NULL);
    } //if (cbTarget<961)
  } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_d (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
  char * pbTargetMax = pbTarget + cbTarget;
  register unsigned long ulHashPattern;
  register unsigned long ulHashTarget;
  signed long count;
  signed long countSTATIC;

  unsigned char SINGLET;
  unsigned long Quadruplet2nd;
  unsigned long Quadruplet3rd;
  unsigned long Quadruplet4th;

  unsigned long  AdvanceHopperGrass;

  long i; //BMH needed
  int a, j;
//Global are next 2 lines already:
  //unsigned int bm_bc[256]; //BMH needed
  //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
  //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
  unsigned long Gulliver; // or unsigned char or unsigned short

  if (cbPattern > cbTarget)
    return(NULL);

  if ( cbPattern<4) {
    pbTarget = pbTarget+cbPattern;
    ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

    if ( cbPattern==3) {
            for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                    if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                }
                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                    return(NULL);
            }
    } else {
    }
    for ( ;; ) {
            if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                return((pbTarget-2));
            if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
            pbTarget++;
            if (pbTarget > pbTargetMax)
                return(NULL);
    }
  } else { //if ( cbPattern<4)
    if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool',

            pbTarget = pbTarget+cbPattern;
            ulHashPattern = *(unsigned long *)(pbPattern);
```

```
        SINGLET = ulHashPattern & 0xFF;
        Quadruplet2nd = SINGLET<<8;
        Quadruplet3rd = SINGLET<<16;
        Quadruplet4th = SINGLET<<24;

        for ( ;; ) {
            AdvanceHopperGrass = 0;
            ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

          if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                count = cbPattern-1;
                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                    if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                    count--;
                }
                if ( count == 0) return((pbTarget-cbPattern));
          } else { // The goal here: to avoid memory accesses by stressing the registers.
                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                    AdvanceHopperGrass++;
                    if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                        AdvanceHopperGrass++;
                        if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                    }
                }
            }

            AdvanceHopperGrass++;

            pbTarget = pbTarget + AdvanceHopperGrass;
            if (pbTarget > pbTargetMax)
                return(NULL);
        }

    } else { //if (cbTarget<961)
        countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
        //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
        //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
        //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

        // Elsiane r.2  [
//Global is next line already:
        //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

        // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
        // fast 4 0-1-2 fa as st
//Global is next line already:
        //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

        // Elsiane r.2 ]

        ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
        //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

        i=0;
  if ( cbPattern>10) { // r.2
        while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
            Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

            if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                    count = countSTATIC; // Last two chars already matched, to be fixed with -2
                    while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                        count--;
                    if ( count == 0) return(pbTarget+i);
                }
                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
    i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
    i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
            } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
            else
                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
```

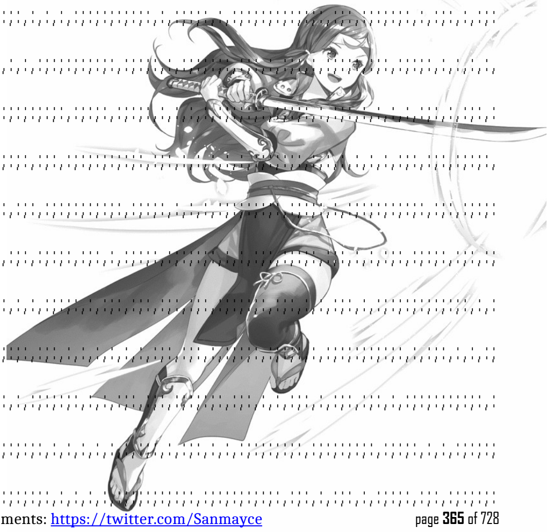
```
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//       longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_e (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
```

```
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
     i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
     i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                        } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        else
                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
```

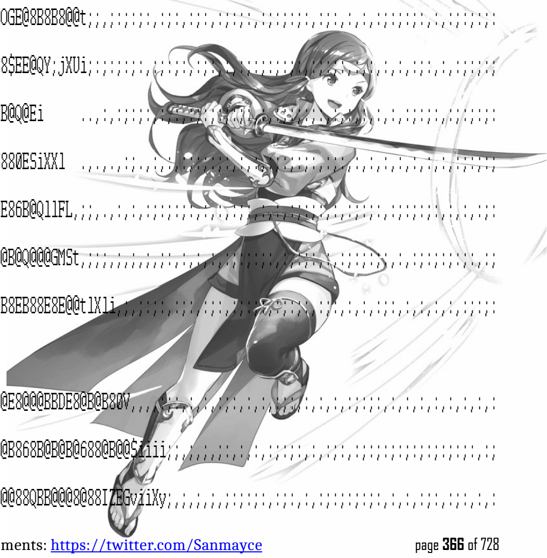
```
// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//       longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                        //GlobalI++;
                }
    } else { // r.2
                while (i <= cbTarget-cbPattern-1) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
    have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                        count--;
                                }
                                if ( count == 0) return(pbTarget+i);
                        }
                        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                        //GlobalI++;
                }
    } // r.2

                if (i == cbTarget-cbPattern) {
                        if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                count = countSTATIC;
                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                        count--;
                                if ( count == 0) return(pbTarget+i);
                        }
                        //GlobalI++;
                }

                return(NULL);
        } //if (cbTarget<961)
    } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]

// Revision: 2, 2012-Jan-30, the main disadvantage: the preprocessing overhead.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
//
char * Railgun_Quadruplet_7Gulliver_f (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    register unsigned long ulHashPattern;
    register unsigned long ulHashTarget;
    signed long count;
    signed long countSTATIC;

    unsigned char SINGLET;
    unsigned long Quadruplet2nd;
    unsigned long Quadruplet3rd;
    unsigned long Quadruplet4th;

    unsigned long  AdvanceHopperGrass;

    long i; //BMH needed
    int a, j;
//Global are next 2 lines already:
    //unsigned int bm_bc[256]; //BMH needed
    //unsigned int bm_bc2nd[256]; //BMS needed
//Global is next line already:
    //unsigned char bm_Horspool_Order2[256*256]; //BMHSS(Elsiane) needed, 'char' limits patterns to 255, if 'long' then table becomes 256KB, grrr.
    unsigned long Gulliver; // or unsigned char or unsigned short

    if (cbPattern > cbTarget)
        return(NULL);

    if ( cbPattern<4) {
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));

        if ( cbPattern==3) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
```

```
                        pbTarget++;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) )
                        return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax)
                        return(NULL);
        }
    } else { //if ( cbPattern<4)
        if (cbTarget<961) { // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.

                pbTarget = pbTarget+cbPattern;
                ulHashPattern = *(unsigned long *)(pbPattern);

                SINGLET = ulHashPattern & 0xFF;
                Quadruplet2nd = SINGLET<<8;
                Quadruplet3rd = SINGLET<<16;
                Quadruplet4th = SINGLET<<24;

                for ( ;; ) {
                        AdvanceHopperGrass = 0;
                        ulHashTarget = *(unsigned long *)(pbTarget-cbPattern);

                    if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
                                count = cbPattern-1;
                                while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
                                        if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                                        count--;
                                }
                                if ( count == 0) return((pbTarget-cbPattern));
                    } else { // The goal here: to avoid memory accesses by stressing the registers.
                                if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
                                        AdvanceHopperGrass++;
                                        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
                                                AdvanceHopperGrass++;
                                                if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
                                        }
                                }
                        }

                        AdvanceHopperGrass++;

                        pbTarget = pbTarget + AdvanceHopperGrass;
                        if (pbTarget > pbTargetMax)
                                return(NULL);
                }

        } else { //if (cbTarget<961)
                countSTATIC = cbPattern-2-2;

//Global are next 3 lines already:
                //for (a=0; a < 256; a++) {bm_bc[a]=cbPattern; bm_bc2nd[a]=cbPattern+1;}
                //for (j=0; j < cbPattern-1; j++) bm_bc[pbPattern[j]]=cbPattern-j-1;
                //for (j=0; j < cbPattern; j++) bm_bc2nd[pbPattern[j]]=cbPattern-j;

                // Elsiane r.2  [
//Global is next line already:
                //for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= cbPattern-1;} // 'memset' if not optimized

                // alfalfa 7 long 6 BBs (al lf fa al lf fa) 3 distinct BBs (al lf fa)
                // fast 4 0-1-2 fa as st
//Global is next line already:
                //for (j=0; j < cbPattern-1; j++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+j)]=j; // Rightmost appearance/position is needed

                // Elsiane r.2 ]

                ulHashPattern = *(unsigned long *)(pbPattern); // First four bytes
                //ulHashTarget = *(unsigned short *)(pbPattern+cbPattern-1-1); // Last two bytes

                i=0;
    if ( cbPattern>10) { // r.2
                while (i <= cbTarget-cbPattern-1) { // -1 because Sunday is used
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];

                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                        count = countSTATIC; // Last two chars already matched, to be fixed with -2
                                        while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                count--;
                                        if ( count == 0) return(pbTarget+i);
                                }
                                //i = i + 1; // r.1, obviuosly this is the worst skip so turning to 'SunHorse': lines below
// r.2 [
```

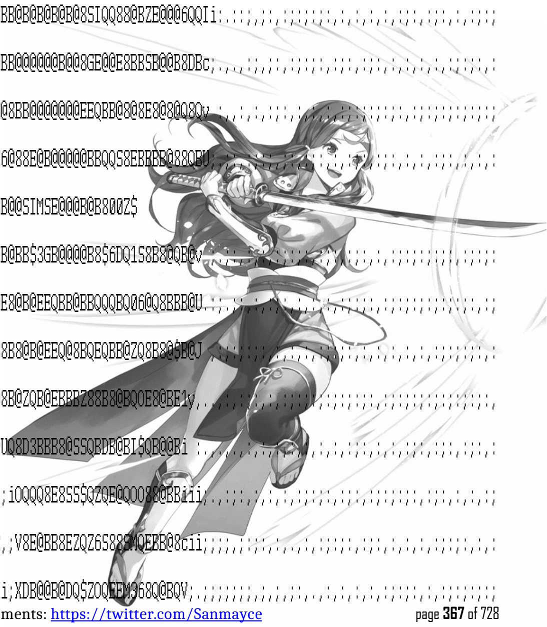
```
if ( bm_bc[(unsigned char)pbTarget[i+cbPattern-1]] < bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]] )
        i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
else
        i= i + bm_bc[(unsigned char)pbTarget[i+cbPattern-1]];
// r.2 ]
                                    } else if ( Gulliver == cbPattern-1 ) // CASE #2: means the pair (char order 2) is not found
                                                i = i + Gulliver; // the pair is not found, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                                    else
                                                i = i +  cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position

// 32323218 Order 1 Horspool
// fa af fa af fa as st Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// fafafast
//   fafafast +2 Order 1 'a' vs 't'
//   fafafast +2 = (cbPattern-Gulliver-2 = 8-4-2 = 2) Order 2 'fa' vs 'st' i.e. CASE #3

// 76543218 Order 1 Horspool
// lo on ng gp pa ac ce Order 2 Horspool
//  0  1  2  3  4  5  6
// HIKARIfast
// longpace
//   longpace +2 Order 1 'a' vs 'e'
//        longpace +7 = (cbPattern-1 = 8-1 = 7) Order 2 'fa' vs 'ce' i.e. CASE #2

                                    //GlobalI++;
                        }
      } else { // r.2
                        while (i <= cbTarget-cbPattern-1) {
                                    if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                count = countSTATIC;
                                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) ) { // if pattern length is 4 or 5 we
      have count=-1 and count=0 respectively i.e. no need of comparing in-between chars.
                                                            count--;
                                                }
                                                if ( count == 0) return(pbTarget+i);
                                    }
                                    i= i + bm_bc2nd[(unsigned char)pbTarget[i+(cbPattern)]];
                                    //GlobalI++;
                        }
      } // r.2

                        if (i == cbTarget-cbPattern) {
                                    if ( *(unsigned long *)&pbTarget[i] == ulHashPattern) {
                                                count = countSTATIC;
                                                while ( count !=0 && *(char *)(pbPattern+(countSTATIC-count)+4) == *(char *)(&pbTarget[i]+(countSTATIC-count)+4) )
                                                            count--;
                                                if ( count == 0) return(pbTarget+i);
                                    }
                                    //GlobalI++;
                        }

                        return(NULL);
            } //if (cbTarget<961)
      } //if ( cbPattern<4)
}
// ### Mix(2in1) of Karp-Rabin & Boyer-Moore-Sunday-Horspool algorithm ]
*/
// Commented 2019-May-17 ]]]]]]]]

/*
RT;.,;;;,;;;;;.;;;;;;;.;.;;;;;;;;;;;.;.;;;.;.;;;.;.;;;.;;;.;;;;;.;;;;;.;.;;;.;.;.;.;;;.;;;;;.;;;.;.;.;.;;;;;.;.;;;;;;;.;;;;;.;.;;;.;;;.;.;;;.;.;;;;;;;;;;;.;;;;;.;.;.;.;;;.;.;;;.;;;.;.;;;.;.;.;.;;;;;;;;;;;.;;;;;.;
      ;.;;;.;.;.;.;.;
.;.,.;.;;;.,.;.;;;.;;;.;.;;;;;;;.;;;.;.;;;.;;;;;.;.;.;.;;;;;;;.;;;;;;;.;;;;;;;;;;;;;.;;;;;;;;;;;.;.;;;;;.;;;;;.;;;;;;;;;.;.;;;;;;;;.;.;.;;;.;;;;;;;;;;;;;.;.;;;.;;;;;.;;;;;;;;;;;;;;;.;.;;;.;;;;;.;.;;;.;;;.;
      ;;.;.;;;;;.;;;.
,.,.;;;;;;;.;.;.;.;.;.;;;.;;;;;.;;;;;;;;;;;.;.;.;;;;;.;.;.;.;;;.;;;;;;;;;.;.;;;;;;;;;.;.;.;.;;;;;.;.;.;.;;;.;;;;;;;;;;;;;.;.;;;.;;;;;;;;;;;;;.;.;;;;;;;;;;;.;.;.;;;;;.;;;.;;;;;;;;;;;;;.;;;;;.;;;;;.;;;.;;;.;
      ;;;;;;;.;;;.;;;
:.;;;;;.;.;;;;;;;;;.;.;.;;;.;;;.;;;;;;;;;.;.;;;.;.;;;.;;;.;.;.;;;.;;;.;.;.;.;;;;;.;.;.;;;;;.;;;;;.;.;;;.;;;;;;;;;.;;;.;.;;;.;;;;;.;.;.;;;;;;;;;.;;;;;;;;;;;;;.;;;;;.;;;;;;;;;;;;;;;.;;;;;.;.;;;.;.;;;;;.;.;;;.;;;
      ;;;;;;.;.;;;.;.
;;;.;;;;;;;.;;;;;;;;;;;;;;;;;;;.;;;;;.;.;;;;;.;;;.;;;;;;;;;;;;;;;.;.;;;;;;;.;.;;;;;;;;;;;;;.;;;;;;;;;;;;;;;;;.;.;;;;;;;;;.;;;;;;;.;.;;;;;;;;;;;.;;;;;;;.;;;;;;;.;;;.;;;;;;;;;;;;;.;.;;;.;;;;;.;;;.;;;.;.;;;.;;;
      ;;;;;.;;;.;.;.;
:;;;;.;;;;;;;.;;;;;.;.;.;.;;;;;.;.;;;;;;;;;.;;;;;;;.;;;.;;;.;.;.;;;;;;;;;;;.;;;;;.;;;;;.;.;;;.;.;;;;;;;;;;;;;;;.;;;;;.;.;;;;;;;.;;;;;;;;;.;.;;;;;;;;;;;;;.;;;;;.;;;;;;;;;;;;;;;.;;;;;.;.;;;;;;;.;;;.;.;;;.;.;;;
      ;;;;;;;;;.;;;;;
;;;;;.;;;.;;;;;;;;;.;;;.;.;.;;;.;;;;;;;;;;;.;;;;;.;;;;;;;;;;;;;;;.;;;;;.;.;;;.;;;;;;;;;;;;;.;;;;;;;.;;;;;;;;;.;;;;;;;;;.;;;;;.;;;;;;;;;;;.;;;;;;;;;.;;;;;;;;;;;;;;;.;;;;;;;;;;;.;;;;;.;;;;;.;.;;;.;.;;;;;;;.;
      ;.;;;;;.;.;.;.;
,.;;;.;;;.;.;.;;;.;;;;;.;.;;;.;;;;;.;.;.;;;;;;;.;;;;;;;.;;;.;.;.;;;;;.;.;.;.;;;.;;;;;.;.;;;.;;;;;;;.;;;;;;;;;;;.;;;;;.;;;;;;;;;;;.;;;;;;;.;.;;;;;;;;;.;;;;;;;;;.;;;;;.;;;;;;;;;;;.;.;;;.;.;;;.;.;.;;;.;.;;;
      ;;;.;.;;;.;;;.
;;.;;;;;.;;;;;;;;;.;;;.;;;;;.;;;;;;;.;;;;;.;;;.;;;;;;;;;;;.;.;;;.;;;;;;;.;;;;;;;.;;;;;;;;;.;.;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;;;;;;;.;;;;;.;;;;;;;;;;;.;;;;;.;;;;;.;;;.;.;;;;;;;.;;;;;
      ;.;;;;;.;.;.;.;
:.;;;;;.;.;.;.;;;;;.;;;;;;;.;;;;;;;;;.;;;;;.;;;;;;;.;;;;;;;.;;;.;.;;;.;.;.;;;;;;;;;.;;;;;.;;;;;;;;;;;;;.;;;;;;;.;;;;;;;;;.;;;;;;;.;;;;;;;;;;;.;;;;;;;;;.;;;;;.;;;;;;;;;;;.;.;;;.;;;;;.;;;.;;;.;;;;;.;;;.;;;
      ;;;;;;;;;.;;;;;
;;;;;.;;;;;;;;;;;;;.;;;;;;;;;.;;;;;.;;;.;.;;;;;.;;;;;;;.;;;;;;;.;;;;;.;;;.;;;;;;;;;;;.;;;;;;;;;;;.;;;;;.;;;;;;;;;.;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;;;;;;;.;;;;;;;;;;;.;;;;;.;;;.;;;.;.;;;.;.;;;;;.;;;;;;;
      ;;;;;;;;.;.;;;
,;;.;.;;;.;.;;;.;;;.;.;;;.;;;;;.;.;;;.;;;;;;;.;;;;;.;;;;;;;;;;;.;;;;;;;.;;;;;;;.;;;;;;;;;;;;;.;;;;;.;;;;;.;.;;;;;;;;;;;;;.;;;;;;;;;;;.;;;.;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;.;;;;;.;;;;;.;;;.;.;;;.;.;;;;;;;;;
      ;;;;.;;;;;;.;.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;.;;;;;;;.;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;.;;;;;;;.;;;;;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;;;;;.;;;;;;;;;.;;;;;;;;;;;;;.;;;;;;;;;;;;;.;;;;;.;;;;;.;;;.;.;;;.;;;;;;;
      ;.;;;;;;;.;.;.;
;;;;;;.;;;.;;;;;;;.;.;.;.;;;;;.;.;.;;;.;.;.;;;.;;;;;;;.;.;;;;;;;;;.;;;;;.;;;.;.;;;;;;;;;;;.;;;;;;;.;.;;;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;;;;;.;;;;;;;;;.;;;;;;;;;;;.;;;;;;;;;;;;;.;;;;;.;;;.;.;.;;;.;.;;;;;;;
```



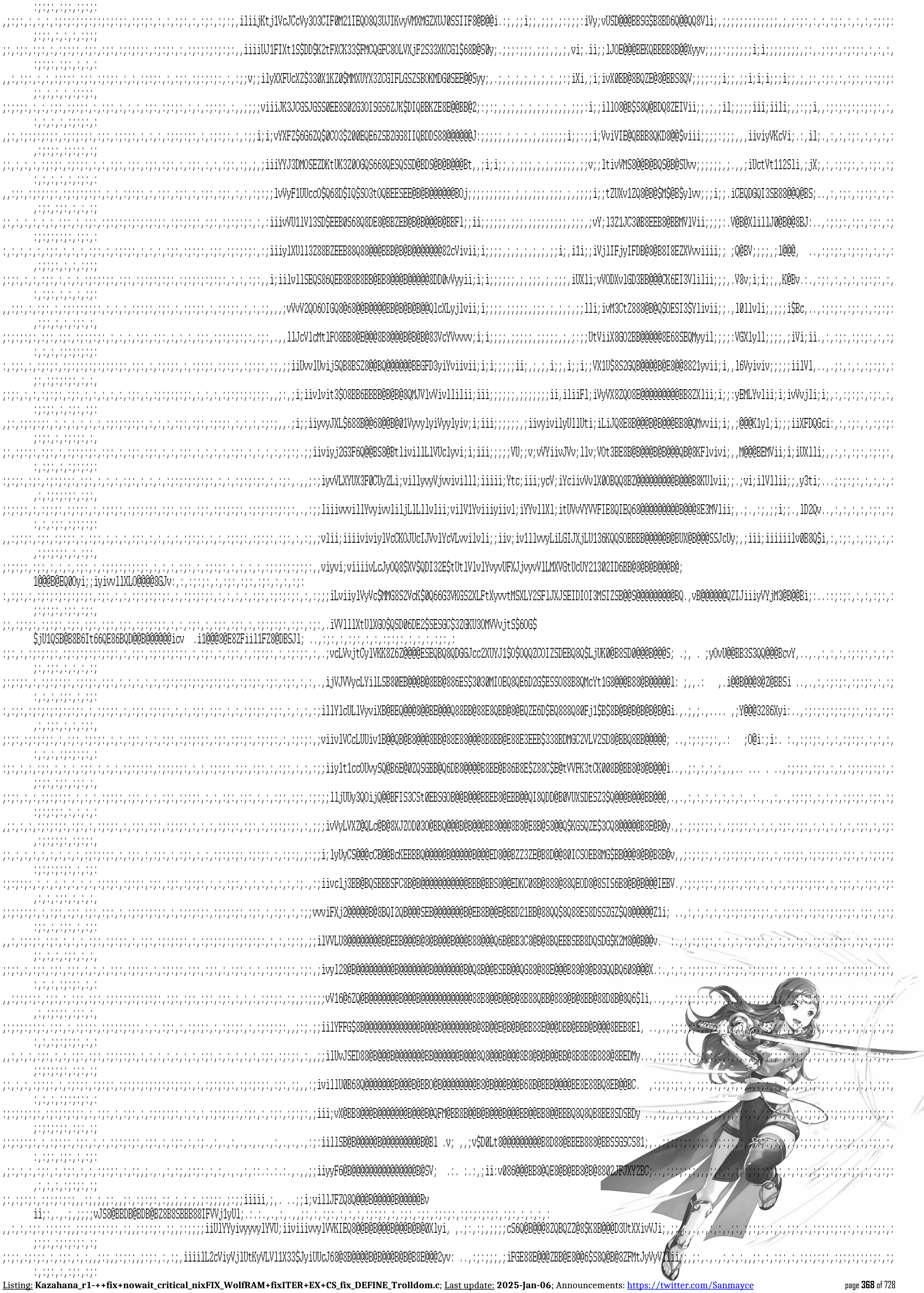
```
       ,:,:;:;:,:;:;:,
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :,:,;:;:,;:,:,:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:,.,.,.i;,.,,.,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:,:;:;:,:;:,:;
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:. ,iyvl;:i; ,.,,.,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :,:,:,:,:;:;:;:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,.,.,.;vOXViiv3VYi,.,;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:;:;:;:,:;:,:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:,.,.,.,ivtQ@DFVlv2K8BL..,.,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :,:;:;:,:,:;:,:
,;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:,.,.. : ..:;ijOBE@B@E@@@Bt;yVi;,:.,.,:,.,.,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:,:;:;:,:;:;:,
;:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,.,:,:,.: :ii;;   ,;ijZS8B@8B@@@@B@ECVylLyi;,,,.,.. ..,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;:;:,:;:,:;:,:
,,:;:;:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,.,.,.,.;i;lBlyEtjOF16BB@EBQQQ@B88BE@8XvUVlivvi;;. ,yi:.,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ,:,:;:;:,:;:,:,
;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,.,.;,;,:.:.iV336ScX88@B@B@EBB@8S1C3MyF8B8SlLUciyvVi;ii;@B@i..,:;:,:,:;:;:,:;:,.,.. :,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :,:;:;:,:,:;:,:
,;:,:,:,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;i;Yl;,iYJljlyt$OEE@@@EQZB@@ED32G0KEG2FX1McCG$XVyvi;i@0;.,.,.,:;:;:;:,:;:;:,:;:, .i; ::;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;,:;:;:;:,:,:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,.;i;yiivl;;;;;yLVlK3IX3D@B@BBB@88Z8@@83KOXIGSICIBBQ1Jly;;;, ::;:,:,,;:;:,;:;:;:,.
       $@@; :,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;it1cyVii;,,;iSE$XMj11tllJ8@@B8@@@Ei.  .O@SGZ@DQQ8SKL1VLvi;,.,.;:;:,:,:,:;:;:,,.
       @B@2, ,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
;:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;ijVJVFUyii;,,;U8BXOUi;iiivEBB@@Q;
       i@B6Q@BBQDI3XFKMGOii;;.,.;:;:;:,:;:;:,. .B@B@; .,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
:,:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;yiVUJcFUYii;;,..y0QZ1ivc0XVC8@@I     ;@@@@@@@@8O6Z$F68@8L;;i,:..:.,,:;:,:,:. ,@@8@l .:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:;:;:;:,:;:,:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,;lvlc3XVl2cYi;,;::.Y@BMK0803CE@@Di   . jB@B@@@DCUS@@B@B@B@MUi;;v;;:,.,:,.:;:;:,.
       0@@X@Z; ,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:
,,:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;,lvUXCFL;iyXVyiiii;;v@@DI88St2E@8F$8KU.,B@@@B@iSi;V68@@@@@E6SU;iii;;,;.,.,:,:;::
       B@8K8@; .,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,
;:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,..iLVYMXLi;;iYJVUlyvv;y8@ZSQ80O2@@@B@B@BUO@@@@@jv1;.;;SB@DB@@0QMVvli;,;;i;;,,.,.,. ;@@GYEBv ::,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :,:;:;:,:;:;:,:;:
;:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;iUFJVYUiiiiiUcJVVvyviv8B8686SKCB@@@B@@@B@@@ytLVY;.,:iY@6IZBFCQSKSKJi;;;;;i;;,,. .@@@FUE@v .:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ,:,:;:;:,:;:,:,
;:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,.,:;lXV2JYiiitVvjFVVvlvv;vS@8@88SOK@B@@@@@@@B@; ;Vi,.,:;;Y3QFS$L1@DDQ8QCyjl;.;;;;i;iilSECt2@@V ,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;:;:,:;:,:;:,:
:,:;:,:,:;:;:,:;:;:;:,:;:;:;:,i;;;i;;;YVylVlll1QFijjYlyiviiv2S@8E8838B@@@@@B@@@U  . ..,:;,;;j3O8@OQQ3$B8@8ZSZLyyv;;;iv3MjVyFSB@v .:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:,:;:;:,:;:,:;
;:,:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;iii;yUUiiivlUXQLvyUiYllililVKDQ3BQ$E@@@@@@@E@8; ,,;:;,;,;vCZ@B@QDKQB@8BB@8DFS31;i;;ijJXVK2@B; ,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;:;:;:;:,:,:,:
,;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:;iLylivvy;illiycLljllvlic0EOOQBZB@@B@B8$S@$ .,;,;,;:;,;,iyZ68B@@QDBEE06Q8OXU$jiii;;;LcXU6@i :.,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:;:;:;:,:;:;:,
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;i;i;vyUlY1BJ, ,;;;ccXUjljlc0@QQSEZ6$StCK3ccM@F .;,;,;,;,;,;vY8B@@B@@@B88QB863D32K1lv;ivjVJI0...,;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :,:;:;:,:;:;:,:
,,:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,.,;;;;iiilcKEB@@@i  ,;ljXLFJtXt2B$O6B6OFJVVvyly13@1 :,;,;,;,;,;,iiU@@@88@BQB@B@@@B8QQQE0IUyilyjUEEi :.,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ,:,:;:;:,:;:;:,
;:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;;;i;ivU$@B@B@;
       ilUjlyUcIF$0tS@@BtO1FXKXJUU2@X .;;;,;,;,,.;;iK@@@SV;,;vYFC8B@B@B@8B8S1tvlUXS@; .;,,.,.,;i,;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;lyFZB@@ti;;iUilyjVCFLtVivLMviiiii;,;M@C ,;;,;,;,;,;;ii;i;;  .:.....;;VE@B@8@BBCXlLF@K.
       Yi;.,;vJc;,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;lvc$BB@@@0QDSIEO2EEili,  :;.. . . ;G@0i ;;;,;,;,;,;;v,.   ,.;.,.:..    ,BB@E6GGUVc8B;
       30i:;;yv;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:,.:
,;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:ilivlFXDB@B@@@B@@@B@BEZ@Qy  ;i.,.,..iZXZ3: ;;;,;,;,;,;i;,;:;:,:,:,.,.,..  ;@@Q1F1XL103;VGL,,iU;  ;,;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:;:;:,:,:;:,:
;:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;vvyL3$@B@B@B@@@@@B@B@SBM; ,;;.;,: i0tv6J..;,;;;,;,;;v,;;;,;,;;;;;,:,..
       y@@@OFX1YVB8My; ,yt;;vv;;,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:
:,:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,.,ililF@B@@@@@B@@@B@@@B@@@i..l;:,;..i6XyXBji,,;,;;;;30;:i;,,;:;,;;;:,..
       iB@I20SCMtE$l;: ;L6M$li:: ..,.,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:
;:;:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:,.;vvJZ@B@B@@@@@@@@@@@B@Bi.VX;.,.:
       vG0VivIG3UUUJU6ZB@@;.;i;,,;,;;,;...l8@tL2$XSBCi;.: ;y30@QLi;,vyVi;.,.,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:
:,:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:;yyJS@@@@@@@@@@@@@@@B@yi8B;:,;:.,UFvii;l1QSEGtJCO@@Qli,,;;:,:;:, ;iEBDjjYUJ$D1i;:;.;U1I@ZSKGMZ6l;Vv;.. :.,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ,:;:;:;:,:;:;:
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;YyyvUO@@@B@@@@@@@B@@@BMX@U:vSv:.:iUyYyjlyi;;;,;.,;M@@B6Vv;;;;;ii1E@EJllyXtGS0i;;;:iXMM@@BSQSS66Vli;.;il;;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;:,:,:,:;:,:;:
,;:,:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,;villMQ@@@@@@@@@@@@@@B@DEBci@Ei;v;,t@$Kli,;:;,;,;.,.vVQB@@@88D@B@E$VviUjX1ZDSyi;;:iMcv8B@@@QEOGE@8B8B8@@t;;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:;:;:;:;:;:,:;
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;ii10BB@@@@@B@@@@@B@BB8@BI;lOJ;UFl;;:;,;;;;;;:.::i;iiUUttXV1cVvcFM1MXO0ZXv;,,;t0;2@@EBBB88$EE@QY;jXUi;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;:,:,:,:;:,:;:
,;:,:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;,,iiicQZB@@@@@@@@@@@@B@B@ZOI@@M,,;;;;;;;;;;;ii;;iMviilililV1J1FG1O1K$EMY;;,;jBviB@B8SQE8B@Q@Ei     ..,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:;:;:,:;:;:;:,
;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;iivlF2G6BB@@@@@@@B@@@@@B8Z@@8;,;;;;;;;;;;;USvvVllIFUl2D$cUYLj1t3233D2Li;,;iEG;U@@@EEEBS880ESiXXl  .,.,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;:,:,:,:;:,:;:
,;:,:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,.,;;iiyLKME8@B@B@@@@@@@8@B@@@@@;.;i;,;;;;;;;;,0BB6@J;i$CGKQBQFcVXFOOO3SMUii;;;tBj.ZB@B@@@@@E86B@QllFL,;;,.,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;:,:,:;:,:;:,:;
;:;:,:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,:,.,:;iVVtI88@@@B@@@B@B8B@B@@@K,,i;;.;,;;;;;,,iB@@@X.;SBU30EB@SGMO2$GSMJvi;i;vCQi;B@B@@@B@@@B@Q@@@GMSt,;;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :,:,:,:,:;:,:;:
;:;:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:,.,;;ivlivvcC8B@@@B@@@B@@@B@@v:i;;;;;;i;;;;;iJ33ii;c8Xy02QQ8OIK2ccjLli;ivviji.i@B@B@QBB@B8EB88E8E@@tlXli,;;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ,:;:;:;:,:,:,:
;:,:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:,:,.,;;i;iiJC8B@@@B@B@B@8@@M,;ii;;;;;i;i;vljyyvyvvvlUBJivVlLJFU1F1iivM2GK3Yi,.
       IB@B@8SS@@B$BB@88DEE@@6l;.. .:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:
,,:;:,:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,;ily1t3I88@@@@@8B8@B@;;ii;;;;;i;iiyVCXUlyvUvLj$BC;i;iiyljUKJVL8B@BB8Ei. :8@@8EB3GQ@@E8@@@BBDE8@B@B80V,,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:;:;:,:,:;:;:,
;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;iylVlYlcKDDB8B88QBB@Y;;i;;;;;iiilccI$ZGSO6DSI0IE2l;i;iivvYyLJB@@B@B@8C;.;@@@D8B8GQ8@B868B@B@B@688@B@@$iiii;,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       :;:;:,:;:,:;:;:
:,:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:,:;iiivvyy1FG088B8@8@Fi;iii;;;i;lVO0QDBB@@@B@E8D$M0li;i;iiviyY3B@@@@@BBQGitB@@DQ@@@QB@@88QBB@@@8@88IZEGviiXy;,,:;:;:,:;:;:;:,:;:;:;:,:;:;:;:,:;:
       ;:,:;:;:;:,:;:
```



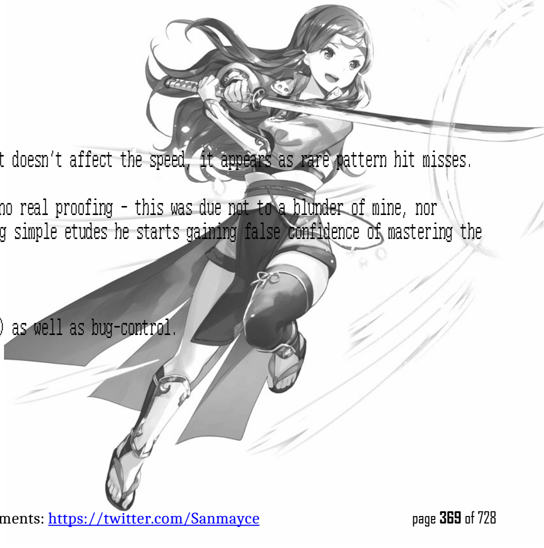
```
;:;:;:,:;:,:;:,:,:;:;:,:,:;:,:;:,:,:;:;:;:;:,:,:;:,:,:iilvliyllvylyJ0DBSy;iii;;;;;lXQQQGD3I0QGS0$S3lcOV;i;iiiilvVS@@@B@@@E8QEB@@@8B@@@D0@BB8B$E@@@@B@8QO6FX;;il;,;,,:,:;:;:;:;:,:,:;:;:;:;:;:;:,:,:;:;:,:;:;:,
        :;:,:,:;:,:,:;:
:,:,:;:,:;:;:;:;:,:;:,:;:,:;:,:;:,:;:,:;:,:;:;:;:,:,,.:.,.;;;;;vJcl;iii;i;;iyM8Q868E6JVii;viv;;lSViiiivivivY8QDBBQB@BB@@@@@B@B@B@ct8@S8BQX8B@@@@@Q88@BEY;   ..,:,:,:,:;:,:;:,:;:;:,:;:;:,:,:;:;:;:,:;:;:;:
        ;:,:,:,:;:;:,:
;:;:;:;:,:;:,:;:;:;:;:,:;:;:;:,:;:;:,:;:;:;:;:;:,:,:,.,.,:;:;:,,:;;iii;vly;;;;;VI888E8QBEt;,:;,;;;,iVcvi;iiiiiicQDtE88DBE@@@@@@@@@B@@FUEB8Z@@DG@B@B@B@@@@@B@B@6i.. :.,.,:,:;:;:;:,:,:,:,:,:,:;:,:;:;:;:
        :,:;:,:;:,:;:;:
,,:;:;:;:;:;:,:,:;:,:;:,:,:,:;:,:;:,:;:,:;:;:,:,:;:;:,:,:,:,:,,:;:;ilJviii;X$BDEZES3MDL;.;;;;;;;;vjVii;iiiiVB8$3S$@Q6@@B@B@@@@@B@Uy3EESC@@EQ@@@@@888@B@B8B@@BLVvi,,,:,:,:;:;:,:;:;:,:;:;:,:;:;:,:,:;:;:;:
        :;:;:,:,:;:;:;:
;:,:,:;:,:,:,:;:,:,:;:;:;:;:;:;:;:;:,:;:;:;:,:;:;:,:;:;:..cjvVcvll1686$G62XvlcU;,,;;i;i;;lli;i;i;iit8@8EFSBB2B@@B@@@B@B@BU;cOSE$Q@@@8@B@B@@@EBB@BB$@8Kvi;;,;,:,:,:,:,:,:;:;:;:;:,:;:,:,:,:,
        :;:;:;:;:;:;:;:
,,:;:;:,:,:;:;:,:;:;:,:,:;:;:,:;:;:,:;:;:;:,:,:;:,:;:,:;yUi,y0OtQ8E6$1MUyvviVli:;;;;;i;;ivii;i;iivLEQ6Q@B@BBB@B@B@B@B@B@Ot26S@BDD@@@@@@@8@B@@@B@B8Q@SCFi ..,.,:;:;:,:;:,:;:,:,:;:,:;:;:;:;:
        ,:;:;:;:,:;:,:
;:;:,:;:;:,:;:;:,:;:;:,:;:;:,:;:,:,:;:;:,:;:;:;:;:,:,.:.LUi,ilMIS6EOIULllvliilV;;,;;;;;;;;i;vii;iiiiLS8OE@@@@@@@@@@@@@@B@@@B@EBE@BBB@@@B@@BQBB@B@B@@@@8$@3i,;:,:;:;:;:;:,:;:;:,:;:,:;:;:,:;:;:,
        :;:;:,:;:;:;:
,,:;:,:;:;:,:;:,:;:,:;:;:,:;:,:,:;:;:,:;:;:,:,:,.:.lU;,vyXUI66KCLVvlvliiiYi;;;;;;;;;;;iivivii;il$8BE8B@B@@@@@B@B@@@@@B@@BB@@@8BB@@@@@BQBBB@@@@@B@B@@@@Gii;;,;,:,:;:;:;:;:;:;:;:,:;:,:
        ;:,:,:;:;:;:,:
;:;:,:;:;:,:;:;:;:;:;:;:,:,:;:;:,:;:;:;:,:,:;:;:,:;:,.ll;,llVtD6SK1YjlYvlii;iii,;,;;;;;;;;i;;;;;iiiil$@B@@@B@B@@@B@B@B@@@B@@@@@@EE8S@@@B@B@QB@BD8B@B@B@BBB@63y;,;:;:;:;:;:,:;:,:,:;:,:,:,:;
        ;:,:;:,:;:;:,:
:,:;:,:,:;:;:;:;:;:,:;:;:;:,:;:,:;:;:;:;:,:;:,:;:,:;il;,iljc00SKFVVllilii;iiyi;;;;;;;;;iii;i;i;;;;i$@@B@@@@@@@@@@@@@B@@@@@B@@@1X8B@@B@B@@@O88EQ@@@B@B@@@@@@QXii;;;;;;;;;;;:;:;:;:;:,:;:;:;:
        ,:;:,:;:;:;:,:
;:;:;:,:;:;:,:;:,:,:;:,:;:;:,:;:,:;:,:,:;:;ii;;viycS$ItXyYlliiiiiiilvi,;;;;;;;;;i;i;iii;i;iV8E8@@B@@@B@@@@@B@B@@@@@B@@@Gi;E8@@@B@@Q68E@@@@@@@B@@@BB8@QUi, :.;:;:;:;:,:;:,:;:;:,:;:,:,:,
        :;:;:,:;:;:;:
,;:,:;:;:;:;:,:;:;:,:,:,:,:,:;:;:,:;:,:;:,:;:;:,:,.,.,;i;;vilVG3OtXVVvliviiiiiivv;;,;;;;;;;;;;;;;;iiiii;yD@D@B@@@B@@@@@@@B@@@@@@@@@@@V, :X@B@E@BB8@B@B@B@B@@@B@8@@It$YV;,.;:,:;:;:;:;:,:;:;:,:;:,:;:
        ;:;:;:;:;:;:;:
;:,:;:,:;:;:,:,:;:;:,:;:,:;:;:,:,:,:;:;:,:,.;;;;vvyVFF3XUyYvlilii;iivili;,;;;;;;;;ivv;;;iii;i;iiG@@8@@@@@B@B@@@@@B@B@@@@@@@@@@v
        t@@@@@@@@@@@@BDBB@@@8B88SSi;ilii;;:;:;:;:;:;:;:;:;:;:;:,:;:;:;:;:;:;:;:
;:;:;:,:;:;:;:;:,:;:;:,:;:,:;:;:;:;:,:;:,:,.,;;;iiiV31CFXyYvlvviiiviiili;,;,;;;;;;;;lvv;;;i;i;i;iyB@8B@@@@@@@@@@@@@@@@@B@@@EIS$;. .8@B@D@@@@@@@8D6BB@BBBBQ@O;, .i;;;;,;:,:;:;:;:;:;:;:;:,:;:
        ,:;:;:;:;:;:;:
;:;:;:,:;:;:,:,:;:;:,:,:;:;:;:,:;:,:;:;:,:,:;i;vvVcXYcYyiliv;iiiiiii;;,;;;;;;;;;;;iii;i;;iii;iG@8B@@@@@@B@@@B@@@@@@@@@@BBU;,i,: ;B@2vyBB@B@@@EB8@@@B@8B@6ii,..,:;:;:;:,:;:;:;:;:;:;:,:;
        :,:;:,:;:;:;:
:;:;:;:,:;:,:;:;:,:,:;:;:;:;:,:,:;:,:,.,.;;iiviVvvilvliviviiiiiiiii;;,;;;;;;;;;;;;i;iii;i;;;i;J@@8@@@@@B@B@B@@@B@B@@@@@@@y:.,.: ,i: ;Z@@@@@@@B@B@@@@@SB@83V:.,:;:;:;:;:,:;:,:;:;:;:
        ,:;:;:;:;:;:
;:,:;:,:,:;:;:;:;:;:,:;:;:,:,:,:,:,:;:,:;:,:;iiilivvliiiiivivii;iiiilii;;,;;;;;;;;;;;;;;;;i;i;i;i;;iB@B@@@@@@@@@@@@@@@@B@B@@@@8, .,.: ..;j@@@B@@@BB8@B@@@@QSDK@E;.,:;:;:;:;:,:;:;:,:;:;:
        :,:;:,:,:;:
,:,:;:,:;:;:;:,:;:;:,:;:;:,:;:;:;:;:,:;:;:,:;:;viivyii;iii;;iiiiiiiilvlii;;,;,;;;;;;;;;;;;i;;;i;i;i;i,iM@B@@@B@@@B@B@@@B@@@@@B@@@Ei.  :..  .2@@@BB@@BE8B@BBB@B8$2Vtyi:,:;:;:;:;:;:;:;:;:,:;:
        ;:,:,:;:;:;:
;:;:;:;:,:,:;:;:;:;:;:;:,:,:;:;:;:,:,:,.,;;;;i;;:,;;;;iii;iiviYyy;;,;;;;;;;;;;;;i;i;;;;;i;i;i;iii;VE@B@@@B@@@@@B@B@@@B@@@@@@@@B@J, . .;I@@8@BE8@88QBBBZ8@@06B;
        iiV;,.;:;:,:;:;:;:;:,:,:;:,:;:;:;:,:
,;:;:,:;:;:;:;:;:,:;:;:;:;:,:,:,:;:;:,.,;;;;;i;;;;,:,;;;;;;i;;;;:,.;;i,;ii;;i;i;;;;;;;;;;i;i;viii8BB8@@@@@@@@@B@@@@@@@@@@@@@@B@8;.:;BB@Q8BQCB@BEB8BBBEB@@BEi. ;i;;:;:;:;:,:;:,:;:,:;:;:
        ;:,:;:;:;:;:
;:;:,:;:;:,:;:;:,:;:;:;:;:,:;:;:;:;:;:;:,.,.:.;V2J00S2S0$CMJUlylliiii;i;;;;,;,;;;;;;;;;;;;;;;.,;;iivi;;iii;i,;6@BBB@B@@@@@@@@@@@B@@@@@@@BZO@BB1I@E0B@@I0B@EB8BB@BB8@88v;.:.,:;:;:;:;:;:;:;:,:;:;:;:,
        :;:;:,:;:;:
,,:;:,:;:;:;:,:;:;:,:;:;:;:;:,:;:;:,:;:,:,:;;;icYVjVlLYUYULXLXXKttLXctUXLUVjyYviii;i;;;i;,.,.,;;;;ili;;i;ii;,J8B8B@@B@@@B@@@@@B@@@B@B@B@BMVIKBB@$6SE6DQ@888BE8B@B@BIv@M;.,:,:;:;:;:,:;:,:;:,:;:;:;:
        ,:;:;:;:;:;:
;:,:;:;:;:;:;:;:,:;:,:;:;:,:;:;:,:;:;:,:,.,iilili;;;;;.,:,.,,;,;,;;;;;;;;iivilivililvyVi.,.:.;;;.,,;;;;iii;;;O0EE@@@B@B@@@@@@@@@@@@@@@@@@X10ESM0IS6EQ@@BQB88DBBBE@I;YU;;;;;,:,:;:;:;:,:,:;:;:,:;:,
        :,:;:,:;:;:;:
,,:,:;:,:;:;:;:;:;:,:;:;:;:,:,:;:;:,:,:,:,iii;i;i;i;;;;;,;,;;;,:,:,.,:,.,:,:;:;;;;;;;;i;;;;;,vCUv;i;;,;;i;i;;UBBB@@QB@@B@@@B@@@@@@@B@B@B@8Q8MKGOZQ@BE8@@BB@8BB@Qcv1K1.:,;:;:;:;:;:,:;:,:;:;:,:
        ;:;:;:;:;:;:
;:;:,:;:;:;:;:,:;:;:,:;:;:;:;:;:;:,:;:;:;:;;;;iiivvii;;;;;;;;;,;;;;;;;;;,;;;,;;;,;;,;,;;;,;;;i;i;lVF2CjYii;i;;;;;iZ@8@@BE@B@@@@@B@@@@@@@B@8B@@B@$O0S8@@B$@B@Q@BBQ@BBtUiiQi ;:;:;:;:,:;:,:;:,:;:,:;:,
        ;:;:;:,:;:;:
:,:;:,:;:;:;:;:;:;:;:,:;:;:;:;:,:;:;:,:,:;;;iYlyvviiiiiiii;;;;;;;;;;;;;;,;,;;;;;;;;;;;;;;;;;;,;,iyt2$cy;i;;;;,GBZMBB88BB@8@@@B@@@@@@@@@@@@@@QQE@8@B@EB@@BZB@DB@@B@Iy.vJ;.;,:,:;:;:;:;:,:;:;:,:;
        :;:;:,:;:;:
;:;:;:;:,:,:;:;:;:;:;:,:;:,:;:;:;:;:,:;:;;;vvYlylYljlyii;iii;i;;iii;;;;;;,;;;;i;;;i;i;;;;;;;;;,il2OXvi;;:;yEX12@DEZE6QDBB@@@@@B@@@@@@@@Q$O@B@8@8BB@@@8@EOB@@@@;iV:;i;.:;:;:;:;:,:;:,:;:;:,:;:;:
        :;:;:;:,:;:
;:,:;:;:;:;:;:;:,:;:;:,:;:;:;:;:;:,:;:;:,;;iililvyljVUlVlylylYlylYlylVlviviviiii;i;i;i;i;i;iii;;;;;;.,,vtGUi;;,ilyiU6EQ$3888E@@@@@@@@@B@@@BB$8B@BB@@8B8@BQB@ZQB@@O ,;:.,:;:;:;:;:,:;:,:;:,:;:;:;:
        ;:,:;:;:;:;:
;:;:;:,:;:;:,:;:;:,:;:;:;:;:,:;:;:;:,:,:,iVVyJc1cF13XFX1cFK$S3U1XMtKMCUUL1tXJX1Yllliiii;ilyii;i;vii;;;;.;l2ci,;;vylvVc8E6Z8ZEB@B@B@@@B@B@@@B8QBB@8@@@8B@8E@0SB@3C1;.,,:,:;:;:;:;:,:;:,:;:;:;:;:
        :,:;:,:;:;:
,,:;:,:;:;:;:;:,:;:;:;:;:,:;:;:;:;:,:,:;ivUYjVXcXc2ttKGFKKG080ItXUXM8DGX1F3C$CXF0XcVLvvlciiilivlYyv;i;i;;vi;;;y$Ui,;2@BE06DBB@B@@@@@B@@@BB6EQB8B@@Q8B@QEEZQ@Elij;.,,:,:;:;:;:,:;:,:;:;:;:;:
        ,:;:;:,:;:;:
,;:;:;:,:;:;:;:,:;:;:,:;:;:;:,:,:,:;:;;vvVU1cXjMCG1KVFCC13cG$I1IO60GcXt0FJUJVXFICMUjYFUl;vviiyvlii;li;;i;i;;;lXSv;ijQ@BEDEEBB@@@@@B@@@@DCQ88Q@B@BSS@@@Q@DEB@V,;i.;:;:;:;:;:,:;:,:;:;:,:;:
        ;:;:;:;:;:;:
;:;:;:,:;:;:;:;:;:;:,:;:;:,:,,;:;:,:;:;vlXtMX1F0M21F2ZtOD2V13$30I21KFS$ClyL2UUjtVcUK1Xlviliiiilvvivvi;iiiii;;iVUyLZtK36BQXQB@@@@@@@@@@@QQ6Q0QB@BB2EB@BQ@BilY; ;;:.;:;:;:,:;:,:;:;:;:,:;:;:;:
        ,:,:;:;:;:;:
;:,:;:;:;:;:,:;:;:;:,:;:;:,:;:,:;:;:;:iiUJ2XCKO1J2D1UMQFUcZ6SF3FtVtFFYyjIXcVccJjXVjvviyvviVlliiii;i;i;iii;iiUltQCi;iQE0Q@BBB@B@B@B@B@8SIQQ88@BZE@@@6QQIi:.:;,:;:;:;:;:,:;:,:;:,:;:;:;:
        ;:,:;:;:;:;:
,;:;:;:,:;:;:;:,:;:;:,:;:;:,:;:;:;:;i;iivV1XXU2O0XZ0KJC1M2SCMXCtJyUcMcLVFFYvylcUK1jviivvyiYVliviv;vyl;i;iil;;UUvKcCli1BBBQ@BB@@@@@@B@@8GE@@E8BBSB@@B8DBc;.,.,:;:;:;:;:,:;:,:;:;:,:;:;:;:
        ;:;:;:;:,:;:
;:;:;:,:;:;:;:;:;:,:;:;:;:,:;:,:;:;ivii;VU3F1XSS0I1J602S0JXyJC$1Jc2FtVUVjyyljVUVUyyvyyjlyyy;iyUiiiyvv;iivv;;lv;;JSMvYOBB8E@8BB@@@@@@@EEQBB@8@8E8@8@Q8Qv :,:,:;:;:;:;:;:;:,:;:;:,:;:;:;:;:;:
        ;:,:;:;:;:;:
,,:;:;:;:;:;:;:,:;:;:,:;:;:,:;:;:;:iiyyUM01F3SJFXIO3cItVyX3SFtUXJFJJlVljjLJCyliyyVlLlliLliijyv;iiiiyilLi:;;;;,c8yvlME@8E6@88E@B@@@@@BBQQS8EBBBB@88QBU;;:;:;:;:;:;:,:;:;:;:,:;:;:;:
        ;:;:;:;:;:;:
;:,:;:;:;:;:,:;:;:;:;:,:;:;:;:;:;:;ivvllttFtIXUU3FJyXFGcUXGcliyVtJXVLlYUOVULYlyvVlllVvllv;vvVllvl;ivyFU:;;;:l33yilOK8QBBB@@SIMSE@@@B@B800Z$
        $@B@@B8@y;:;:;:;:;:;:;:;:;:,:;:;:;:;:;:;:;:;:;:,:;:;:;:
;:,:;:;:;:;:;:;:,:;:;:;:;:;:;:,:;:;:;i;YJXL3FLyXXMUJLtcFXFUYvUU1KXvllliX0XlYljlVVLvyyyiliiiVVULUvVlyLV.,;i,;V8i..;vVGBSEB@BB$3GB@@@B8$6DQ1S8B8@QB@v;:;:;:;:;:;:,:;:;:;:,:;:;:;:;:;:;:;:;:
        ,:;:;:;:;:;:
,;:;:;:,:;:;:;:;:,:;:;:;:;:,:;:;:,:,;iivYtjctIUCccjMKtYcU1UUVX3GXKYviVvvy1yjccYYvjUYvYvlvyljYliyyVyXtG;.,i;;,ii,.lyyltZ8OE8@B@EEQBB@BBQQQBQ06@Q8BBB@U.:;:;:;:;:;:;:,:;:;:;:,:;:;:;:;:
        ;:,:;:;:;:
;:,:;:;:;:;:,:;:;:;:;:,:;:;:;:,:,.,;;;;iiyVXUXXKjJjtt3MGFtYylJJXVUVLjXLjyXyvjScjlvVFlllUXtVyyUlliV1XJ8L,.;;;;;;;;ycVlXvcQ88B8@B@EEQ@8BQEQBB@ZQ8B8@$B@J ;:;:;:;:;:,:;:;:;:,:;:;:;:;:
        ;:;:;:;:;:;:
;:,:;:;:;:;:,:;:;:;:;:,:;:;:;:,:,,;;;;iijXXVJUyL$O3UJc3FjYMcUyjLUF0tJyVyyiYVllYiLjYvYYCGJilt3LXK1F8V;.,;;;;,,.vUY;;Vv;3O8B@ZQB@EBBBZ88B8@BQOE8@BE1y,.;:;:;:;:;:;:,:;:;:;:;:;:,:;:;:;:
        ;:;:;:;:;:;:
,;:;:;:,:;:;:;:;:,:;:;:;:;:,:;:;:;:;i;;;lVXUJVUUcVJU1VyytVFILvVcFlUylYMyvivilvcjllliylyVMMCUUlVF01EG;.,,;,;,;;;ii;i;;;vivUQ8D3BBB8@SSQBDB@BI$QB@@Bi :.,:;:;:;:;:,:;:;:;:,:;:;:;:
        ;:;:;:;:,:;:
;:,:;:;:;:;:;:;:,:;:;:;:;:;:;:,:;:;:;iiYycJKVjYUjUVUV1JLVJYUJ3VvVUvMMlililVccjycyVc3XCUt0CytI3cBE;.,,;,;,;,;;ly;,;;;,v;;iOQQQ8E8SS$QZQE@QOO88@BBiii;.,:;:;:;:;:;:,:;:;:;:,:;:;:;:
        ;:;:;:;:;:;:
;:,:;:;:;:;:;:;:;:,:;:;:;:,:;:;:;:;:;iiiiyVF1MVXccjcU13Jivi3OZFlvUUJCFvVVYyUyViYXXj31UVUU32IXCOBBK:::;,;;;,;;;;J;;;i;iS2,;V8E@BB8EZQZ6S88SMQEBB@8cii;;;;;:;:;:;:;:,:;:;:;:,:;:;:;:
        ;:;:;:;:;:;:
;:;:;:;:,:;:;:;:;:;:;:,:;:;:;:;:;:;:;il;ilclyjcjXUcYLj3YlvY6BZMV0cXG0cJVKVYyUJVJOKXjUyJK$SQCXCBBQ,.:;;;,;,;;;;,;i;.;,;1@Xi;XDB@@B@DQ$ZOQEEM368Q@BQV;.;;;;;;;;,:;:;:;:,:;:;:;:;:;:;:
```



Listing: **Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_Trolldom.c**; Last update: **2025-Jan-06**; Announcements: https://twitter.com/Sanmayce



```
,,:;:,:;:,:;:;:;,;:;,;:;:;:;:;:,:;,;;iiviiivii;;vlliyj1F$KCLGCSQQD@@B0IO6GSE@@@@@@@@B8@B@EB8@EEZCcM8@@@X;...,:;:;:;;vVBBB88B@BB@@8B@@BEG8Q88@@@EDG63FlyljVliliiivii,,:,.,:;:,:;:,:,:,:,:,:;:,:;:,:,:;:;:;:,:
        ;:;:;:;:;:;:;:;
;:,:,:;:,:,:;:,;:;,;:;,;:;:,.,:,,;;iiyyVivvi;;;iii;lYUyUVFXXXtCO3S08B@@BQ@@@8BB@B@@@@@@@BBS$S@B8Q@SG2D8@@@@X;,.,:,,;;i;lQ@B@B@@@@@B@Q@@@ZSZEO6863B8888OJvllLUUcCUYvliiiii;;,,.,,;:,.:.,:,:,:;:,:,:;:;:,:;:;:;:,
        ;:;:;:;:;:;:;:;
,,:,:,:,:;:,:;:,;:;:;:,;.:.:;;;ivivilvlii;;;;;ylLYlvyvviyyUyVlcLcX3GE8BEE8@B@@@@@B@B@B@B@BQ068BSMX8E0LUM8@@Ey;,:;:;:,..;EB@B@B@88ME8BB@@@B@6S3@8FM@QStcVLjXJ2FKUUlyyYvYYVvi;;;i;;,;...,.,.;:;:;:;:,:,:,:,:;:,:,:
        ;:;:;:;:;:;:;:;
;:,:,:,:;:,:,:,:,:,:,:, :,vyVlllUlyvlilivvylVvvivivivvyvlvyivlLlVU3C$SEQE686EQB8@@@8BE8SQSEEBE86ZXV0SctX1U36SUl,;:;:;,:;I@@8@QZLcUlvtS$USQ8B@BS28UjQSFG30COKFLUYjyliyvVVVvVlylViiii;;iFJ; :.,:;:,:,:,:,:,:;:;:;:,
        ;:;:;:;:;:;:;:;
,;:,:;:,:;:,:,:,:,:,:,.:V02UlUcUlyyVvvillcLVvYlliviiillllvllvvUccVcXMK3FCXFC$I$IZZ8EBZSFFXMM6B@E88@EBSZQ8I1JMMOl;.;,;,;,.,ij;;;ii,;li;;v;ili;lS@U;;;iXv1E8F13$yllccXlVVcVjlyyUylvlvyljjMCEJv;, ....:.,.,:,:,:;:;:,:
        ;:;:;:;:;:;:;:;
;:,:;:,:;:,:,:,:,:,:,, ;I@VlilFKVjyVllljyYyUyyycvylyvyvylYycYVyULUlLcCM3XXUKF3CSGSODSZ3GF2MO3E8@B@@@@@B@@@B@SVly;;,;;;,;.,;;.;;;.,;;;i;;iXlYlMDUillviV3IJyilljKFyVUXLJjXYUVLyUjUVLyjyVyUcG2MLl,,.. . ..:.,:,:,:,:;
        ;:;:;:;:;:;:;:;
:,:;:;,:;:,:,:,:,:,:,.;I8Jv,ijKVcXCLi;ylyvylVvylyyVlYlYlLVUYVvllJVVlUXIM3F3M3C0GZO$3$GD6QO$33KSQ@BB8BB@B@B@B@F. ;;;;;;;;;;ivyii;;;;;;ii;iiVLVivijVVvj1MlVLyi;iZIFJ3tcylilyyiiilyjVJLcVYvllLlX3KJXYyii:. . ..,.,:,:
        ,;:;:;:;:;:;:;:;
;:;:,:,:,;,;,;,;,;,;,;,;:
        c@BFi;y0UcXFKi;ycyvlvyivivvlljyVyVYjYjvlvjjVlVUFF3F3K212FGISG$M0SQ$Z$6KG0EEBB@8@B@B@B@B@V;ivivii;iicK2VY;;;i;v;;;i;;,;;viv;;;i;vyjKFlcXGS6GGUyvYllvVlYvlvlvYyUUtX1XKtMXKMIF31G3ILVii,. . ..,:;,;,;:;,:;:
        ;:;:
,,:;:;:;:,:,:,:;:..;B@B8$MES1VUV1yC8BZSYlivilvYlVlVyVlYvVUUVcctVVYVlVYJJ11GC0GZ06G0I$IG2S$EQB8@B@@@@@@@B@@@B@;ilVljlyvVXCJUYvii;lvli;;;;;,ivcLliv;;il;VX3FQS0cUjXX$68SQ68SZZQF3C$0QSQZ60DSQGIcUlylXXJX0tcVVi;...
        ;:;:;:;:;:;:;:;
;:,:,:;:;:,:,:;:,,i@@@@@BBQQMXLLJ6DSQZt8LvillylVvlvYlVlyljYUVcVLllillXMS6B8B8@BB8BB@886ED88BB@B@B@@@@@B@@@@@ZilKlLXcyUX$MXyViVlvyVvjvvii;;;Y12llviiV;,;vlvilCIcMGGLjcCKIF1X0S6SEZEQ8E@B8QB@@B@B@8QS88ESQ$
        $0EEBX;.;:,:;:;:;:,:,:
,,:;:;:,;,:,:,:,OB@@@B8ZEB@0OQBS8KXyU8QylvLyyvyvlvlvlvlilvlvvilyM6@B@@@B@@@@@@@B@@@@@B@B@B@@@B@B@B@@@B@B@@OU0KcUtcLL2I2yYVcVUlylviVVYiviy2ZVlvliiiiivvVi:
        llj$@Qi                 . .,;;;;iiiiilljYX2D0E@@@@@@QV.,:;,;:;:;:,:,:;
;:,:,:;:;:,:;:,:,;;;;v@@@@@@@@@Q3S2jQB@8ZtMG$CMXKXtLJVcyvivillJK@@@@@B@@@B@@@@@@@@@B@@@B@B@B@@@B@@@@@@@@@B@B@@B8B68S62CVcUylVV1LMjVyVvlyGQ8XylUli;vYliYVV;iyviyY$8Si. ..:.:.,.:.. . . . .         . . ..:.,.. ,
        ;:;:;:;:;:;:;:;
:,:,:,:;:,:;:,:,:..   ;6@BE8@B@@SF20@@@@@8ESQQ8$SS6OZ0SO$KGIQB@@@@@B@@@@@@@@@B8SZGG2IKMctjlvlvlilvllVVUjKCSDBB@@@@@@BZSKKccVYvylYlUvv;iivC@$Y;ilyivlliviiiUMFyl;;i1EQXy,,;,;,;:;,;:,:,:,.;.,.,.,.:.:.. . . ..,.
        ;:;:;:;:;:;:;:;
;:,:;:,:,.,:;:,:;:,.:..     ...V@@@@@@@@@@@@B@B@@@@@B@@@B@@@B@@@@@BQtVii;;:,.:.:.:.: . . . . ..,.,.;,;,;;;;iilY1K$
        $8EEI3t1cLlyvlivvliiiiiOBZilvyvlilii;iiiiMUVi;i;.;Vti,.;,;:;,;:;,;:;,;:,:,:,:,:,:;:,:,.;:,:,;,:;:;:;
:,:;:,:;:,:;:,:,:,:,.. . . .i@B@BBQ88@@@@@B@B@B@@@8BQ8EQ1Lvi;, . . ..:.,.,.;.,:;:;:,:;:,;:;:;:;:;:;:;:;:;:;:;i;i;iii;ivM$
        $t1Jclliliv;iililvl2@Iylyvlii;;;;iliij0v;;i;,.ilv::.;,;,;:;,;:;,;:;,;:;:;,;:;:;,;:;,;:;:;:;:;:;:;:;:;
;:,:;:;:;:,:;:;:;:;:,:,:,:..                                  ..,.;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;i;;;i;i;i;i;;VSCtUMUVviii;;,;iyYYvOB6jVllvviviiiiiyvy1Liyvvii;JKv:::;,;,;:;,;:;,;:;:;,;,;:;,;,;:,:;:;:;,;:;
        ;:;:;:;:;:;:;:;
,;:,:;:,:;:,:;:;:;:;:;:,;,.. . . . . . . . . . . . . ....:.;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;i;;;i;iiii2ZOXJUcvlljlyi;,;illl3@8IUUYVvliv;iilyVlS0UvYlVvVyFSj.::;,;,;:;,;:;,;:;,;:;:;,;:;,;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;i;iiyltZE3IjLX$C$DEZ6Ji;iillZ@@QItXyVlvii;llylVQ8KFjYvYlllLcl.,;,;:;,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;iilvUZ@8D33CE8BB@S6QQCFi;iVS@BQM3cjyVvi;vljlyl8BQ2tllvvlViYUi.,;,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;i;iilYB@@E8S8B@@@B$F6SEQMlVjE@B2GIMYVlliiilycYUB@StYyii;iiiiLl,.;,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;;iilC@@@B@8@@@@8cX$88@8608B@B0IQ3MM$CIFC1UlyU@@8jliVVlvyiilV;,:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:
        ;:;:;:;:;:;:;:;
;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;;iic8@B@@@@@@@61JD8@BBE@B@@@SQSSS@@@IG6B$XUI@@@SXK6BEEKl;jL;.,;,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;,:,
        ;:;:;:;:;:;:;:;
,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;;;l3B@@B@@@B$1D@@@@B@@@B@@BBBQB@@BtySBBG@@@B@QQ8@BO2E2F3j.,;,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:,:.: .:;iG@@@@B@@@@@B@B@B@@@@@B@@@@@FC8@@@@@@@@@8@@@1$QB@I:::;,;,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;,
        ;:;:;:;:;:;:;:;
,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:,.:..   .;K@@0Y;;..     .;VMBB@B@@@@@B@MXXB@@@@@@@@B@@I:.:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:,.,..          . . .      :iV@BJ;;      .;iQ@@B0Jy...;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:,:,:,.,.,.,:,.,:..          ... .       .   ..;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:,.. . ..;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
,;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;
        ;:;:;:;:;:;:;:;
*/
// Railgun_Trolldom (the successor of Railgun_Swampshine_BailOut - avoiding second pattern comparison in BMH2 and pseudo-BMH4), copyleft 2016-Aug-19, Kaze.
// Railgun_Swampshine_BailOut, copyleft 2016-Aug-10, Kaze.
// Internet "home" page: http://www.codeproject.com/Articles/250566/Fastest-strstr-like-function-in-C
// My homepage (homeserver, often down): http://www.sanmayce.com/Railgun/
/*
!!!!!!!!!!!!!!!!!!!!!!!! BENCHMARKING GNU's memmem vs Railgun !!!!!!!!!!!!!!!!!!!!!!!! [
Add-on: 2016-Aug-22

Two things.

First, the fix from the last time was buggy, my apologies, now fixed, quite embarrassing since it is a simple left/right boundary check. It doesn't affect the speed, it appears as rare pattern hit misses.
Since I don't believe in saying "sorry" but in making things right, here my attempt to further disgrace my amateurish work follows:
Two years ago, I didn't pay due attention to adding 'Swampwalker' heuristic to the Railgun_Ennearch, I mean, only quick test was done and no real proofing - this was due not to a blunder of mine, nor
        carelessness, but overconfidence in my ability to write "on the fly". Stupid, indeed, however, when a coder gets momentum in writing simple etudes he starts gaining false confidence of mastering the
        subject, not good for sure!
Hopefully, other coders will learn to avoid such full of neglect style.

Second, wanted to present the heaviest testbed for search i.e. memmem() functions: it benefits the benchmarking (speed in real application) as well as bug-control.

The benchmark is downloadable at my INTERNET drive:
https://1drv.ms/u/s!AmWWFXGMzDmEglwjlUtnMJrfhosK

The speed showdown has three facets:
- compares the 64bit code generated from GCC 5.10 versus Intel 15.0 compilers;
- compares four types of datasets - search speed through English texts versus genome ACGT-type data versus binary versus UTF8;
```



- compares the tweaked Two-Way algorithm (implemented by Eric Blake) and adopted by GLIBC as memmem() versus my Railgun_Swampshine.

Note1: The GLIBC memmem() was taken from latest (2016-08-05) glibc 2.24 tar:
https://www.gnu.org/software/libc/
Note2: Eric Blake says that he enhanced the linearity of Two-Way by adding some sublinear paths, well, Railgun is all about sublinearity, so feel free to experiment with your own testfiles (worst-case-scenarios), just make such a file feed the compressor with it, then we will see how the LINEAR Two-Way behaves versus Railgun_Swampshine.
Note3: Just copy-and-paste 'Railgun_Swampshine' or 'Railgun_Ennearch' from the benchmark's source.

So the result on Core 2 Q9550s @2.83GHz DDR2 @666MHz / i5-2430M @3.00GHz DDR3 @666MHz:

```
----------------------------------------------------------------------------------------------------------------------
¦ Searcher                                 ¦ GNU/GLIBC memmem()        ¦ Railgun_Swampshine       ¦ Railgun_Trolldom          ¦
|--------------------------------------------------------------------------------------------|-------------------------|
¦ Testfile\Compiler                        ¦ Intel 15.0 ¦ GCC 5.10     ¦ Intel 15.0 ¦ GCC 5.10    ¦ Intel 15.0  ¦ GCC 5.10    ¦
|----------------------------------------------------------------------------------------------------------------------|
¦ Size: 27,703 bytes                       ¦     4506/- ¦   5330/14725 ¦     13198/- ¦ 11581/15171 ¦ 19105/22449 ¦ 15493/21642 ¦
¦ Name: An_Interview_with_Carlos_Castaneda.TXT                         ¦             ¦             ¦             ¦             ¦
¦ LATENCY-WISE: Number of 'memmem()' Invocations: 308,062              ¦             ¦             ¦             ¦             ¦
¦ THROUGHPUT-WISE: Number of Total bytes Traversed: 3,242,492,648      ¦             ¦             ¦             ¦             ¦
|----------------------------------------------------------------------------------------------------------------------|
¦ Size: 2,347,772 bytes                    ¦      190/- ¦      226/244 ¦      1654/- ¦   1729/1806 ¦   1794/1822 ¦   1743/1809 ¦
¦ Name: Gutenberg_EBook_Don_Quixote_996_(ANSI).txt                     ¦             ¦             ¦             ¦             ¦
¦ LATENCY-WISE: Number of 'memmem()' Invocations: 14,316,954           ¦             ¦             ¦             ¦             ¦
¦ THROUGHPUT-WISE: Number of Total bytes Traversed: 6,663,594,719,173  ¦             ¦             ¦             ¦             ¦
|----------------------------------------------------------------------------------------------------------------------|
¦ Size: 899,425 bytes                      ¦      582/- ¦      760/816 ¦      3094/- ¦   2898/3088 ¦   3255/3289 ¦   2915/3322 ¦
¦ Name: Gutenberg_EBook_Dokoe_by_Hakucho_Masamune_(Japanese_UTF8).txt  ¦             ¦             ¦             ¦             ¦
¦ LATENCY-WISE: Number of 'memmem()' Invocations: 3,465,806            ¦             ¦             ¦             ¦             ¦
¦ THROUGHPUT-WISE: Number of Total bytes Traversed: 848,276,034,315    ¦             ¦             ¦             ¦             ¦
|----------------------------------------------------------------------------------------------------------------------|
¦ Size: 4,487,433 bytes                    ¦      104/- ¦      109/116 ¦       445/- ¦     458/417 ¦     450/411 ¦     467/425 ¦
¦ Name: Dragonfly_genome_shotgun_sequence_(ACGT_alphabet).fasta        ¦             ¦             ¦             ¦             ¦
¦ LATENCY-WISE: Number of 'memmem()' Invocations: 20,540,375           ¦             ¦             ¦             ¦             ¦
¦ THROUGHPUT-WISE: Number of Total bytes Traversed: 13,592,530,857,131 ¦             ¦             ¦             ¦             ¦
|----------------------------------------------------------------------------------------------------------------------|
¦ Size: 954,035 bytes                      ¦       99/- ¦      144/144 ¦       629/- ¦     580/682 ¦     634/807 ¦     585/725 ¦
¦ Name: LAOTZU_Wu_Wei_(BINARY).pdf                                     ¦             ¦             ¦             ¦             ¦
¦ LATENCY-WISE: Number of 'memmem()' Invocations: 27,594,933           ¦             ¦             ¦             ¦             ¦
¦ THROUGHPUT-WISE: Number of Total bytes Traversed: 8,702,455,122,519  ¦             ¦             ¦             ¦             ¦
|----------------------------------------------------------------------------------------------------------------------|
¦ Size: 15,583,440 bytes                   ¦        -/- ¦          -/- ¦         -/- ¦     663/771 ¦     675/778 ¦     663/757 ¦
¦ Name: Arabian_Nights_complete.html                                   ¦             ¦             ¦             ¦             ¦
¦ LATENCY-WISE: Number of 'memmem()' Invocations: 72,313,262           ¦             ¦             ¦             ¦             ¦
¦ THROUGHPUT-WISE: Number of Total bytes Traversed: 105,631,163,854,099 ¦            ¦             ¦             ¦             ¦
----------------------------------------------------------------------------------------------------------------------
```

Note0: Railgun_Trolldom is slightly faster (both with Intel & GCC) than Railgun_Swampshine, this is mostly due to adding a bitwise BMH order 2 (8KB table overhead instead of 64KB) path - for haystacks <77777 bytes long - the warm-up is faster.
Note1: The numbers represent the rate (bytes/s) at which patterns/needles 4,5,6,8,9,10,12,13,14,16,17,18,24 bytes long are memmemed into 4KB, 256KB, 1MB, 256MB long haystacks.
in fact, these numbers are the compression speed using LZSS and memmem() as matchfinder.
Note2: The Arabian Nights is downloadable at:
https://ebooks.adelaide.edu.au/b/burton/richard/b97b/complete.html
Note3: On i5-2430M, TW is catching up since this CPU crunches instructions faster while the RAM speed is almost the same, Railgun suffers from the slow RAM fetches - the prefetcher and such suck.
Note4: With a simple English text 'Tales of 1001 Nights', 15,583,440 bytes long, the cumulative size of traversed haystack data is nearly 100TB, 105,631,163,854,099 ~ 1024^4 = 1,099,511,627,776.
Note5: With a simple French text 'Agatha_Christie_85-ebooks_(French)_TXT.tar', 32,007,168 bytes long, the cumulative size of traversed haystack data is nearly 200TB ~ 234,427,099,834,376.

Just to see how faster is Yann's Zstd in decompression (its level 12 is 377-331 MB/s faster), on Core 2 Q9550s @2.83GHz DDR2 @666MHz:

```
[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
D:\Nakamichi_Kintaro++_source_executables_64bit_(GCC510-vs-Intel150)_(TW-vs-RG)_BENCHMARK>Nakamichi_Kintaro++_Intel_15.0_64bit.exe Agatha_Christie_85-ebooks_(French)_TXT.tar
Nakamichi 'Kintaro++', written by Kaze, based on Nobuo Ito's LZSS source, babealicious suggestion by m^2 enforced, muffinesque suggestion by Jim Dempsey enforced.
Note1: This compile can handle files up to 1711MB.
Note2: The matchfinder/memmem() is Railgun_Trolldom.
Current priority class is HIGH_PRIORITY_CLASS.
Compressing 32007168 bytes ...
¦; Each rotation means 64KB are encoded; Done 100%; Compression Ratio: 3.53:1
NumberOfFullLiterals (lower-the-better): 164
NumberOfFlushLiteralsHeuristic (bigger-the-better): 184323
Legend: WindowSizes: 1/2/3/4=Tiny/Short/Medium/Long
NumberOf(Tiny)Matches[Short]Window (4)[2]: 226869
NumberOf(Short)Matches[Short]Window (8)[2]: 119810
NumberOf(Medium)Matches[Short]Window (12)[2]: 71202
NumberOf(Long)Matches[Short]Window (16)[2]: 31955
NumberOf(MaxLong)Matches[Short]Window (24)[2]: 7078
NumberOf(Tiny)Matches[Medium]Window (5)[3]: 257313
NumberOf(Short)Matches[Medium]Window (9)[3]: 526493
NumberOf(Medium)Matches[Medium]Window (13)[3]: 285579
NumberOf(Long)Matches[Medium]Window (17)[3]: 158873
NumberOf(MaxLong)Matches[Medium]Window (24)[3]: 51276
NumberOf(Tiny)Matches[Long]Window (6)[4]: 41075
NumberOf(Short)Matches[Long]Window (10)[4]: 240454
NumberOf(Medium)Matches[Long]Window (14)[4]: 258653
NumberOf(Long)Matches[Long]Window (18)[4]: 209007
NumberOf(MaxLong)Matches[Long]Window (24)[4]: 190929
RAM-to-RAM performance: 605 bytes/s.
Compressed to 9076876 bytes.
LATENCY-WISE: Number of 'memmem()' Invocations: 102,091,852
THROUGHPUT-WISE: Number of Total bytes Traversed: 234,427,099,834,376

D:\Nakamichi_Kintaro++_source_executables_64bit_(GCC510-vs-Intel150)_(TW-vs-RG)_BENCHMARK>"Nakamichi_Kintaro++_Intel_15.0_64bit.exe" "Agatha_Christie_85-ebooks_(French)_TXT.tar.Nakamichi"
Nakamichi 'Kintaro++', written by Kaze, based on Nobuo Ito's LZSS source, babealicious suggestion by m^2 enforced, muffinesque suggestion by Jim Dempsey enforced.
```

```
Note1: This compile can handle files up to 1711MB.
Note2: The matchfinder/memmem() is Railgun_Trolldom.
Current priority class is HIGH_PRIORITY_CLASS.
Decompressing 9076876 bytes ...
RAM-to-RAM performance: 331 MB/s.
Compression Ratio (bigger-the-better): 3.53:1

D:\Nakamichi_Kintaro++_source_executables_64bit_(GCC510-vs-Intel150)_(TW-vs-RG)_BENCHMARK>zstd-windows-v0.8.1_win64.exe -h
*** zstd command line interface 64-bits v0.8.1, by Yann Collet ***
...

D:\Nakamichi_Kintaro++_source_executables_64bit_(GCC510-vs-Intel150)_(TW-vs-RG)_BENCHMARK>zstd-windows-v0.8.1_win64.exe -b12 "Agatha_Christie_85-ebooks_(French)_TXT.tar"
12#_(French)_TXT.tar :  32007168 ->   8965791 (3.570),   6.7 MB/s , 377.0 MB/s

D:\Nakamichi_Kintaro++_source_executables_64bit_(GCC510-vs-Intel150)_(TW-vs-RG)_BENCHMARK>zstd-windows-v0.8.1_win64.exe -b22 "Agatha_Christie_85-ebooks_(French)_TXT.tar"
22#_(French)_TXT.tar :  32007168 ->   6802321 (4.705),   1.0 MB/s , 260.7 MB/s

D:\Nakamichi_Kintaro++_source_executables_64bit_(GCC510-vs-Intel150)_(TW-vs-RG)_BENCHMARK>
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

Two-Way is significantly slower than BMH Order 2, the speed-down is in range:
- for TEXTUAL ANSI alphabets: 1729/226= 7.6x
- for TEXTUAL UTF8 alphabets: 2898/760= 3.8x
- for TEXTUAL ACGT alphabets:  458/109= 4.2x
- for BINARY-esque alphabets:  580/144= 4.0x

For faster RAM, than mine @666MHz, and for haystacks multimegabytes long, the speedup goes beyond 8x.

The benchmark shows the real behavior (both latency and raw speed) of the memmem variants, I added also the Thierry Lecroq's Two-Way implementation:
http://www-igm.univ-mlv.fr/~lecroq/string/node26.html#SECTION00260
However, Eric Blake's one is faster, so it was chosen for the speed showdown.

Once I measured the total length of traversed haystacks, and for files 100+MB long, it went ... quintillion of bytes i.e. petabytes - good torture it is.

!!!!!!!!!!!!!!!!!!!!!!!! BENCHMARKING GNU's memmem vs Railgun !!!!!!!!!!!!!!!!!!!!!!!! ]
*/
// 2014-Apr-27: The nasty SIGNED/UNSIGNED bug in 'Swampshines' which I illustrated several months ago in my fuzzy search article now is fixed here too:
/*
The bug is this (the variables 'i' and 'PRIMALposition' are uint32_t):
Next line assumes -19 >= 0 is true:
if ( (i-(PRIMALposition-1)) >= 0) printf ("THE NASTY BUG AGAIN: %d >= 0\n", i-(PRIMALposition-1));
Next line assumes -19 >= 0 is false:
if ( (signed int)(i-(PRIMALposition-1)) >= 0) printf ("THE NASTY BUG AGAIN: %d >= 0\n", i-(PRIMALposition-1));
And the actual fix:
...
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
        // The line below is BUGGY:
        //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
        // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
        //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
        if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
                if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                                count = PRIMALlengthCANDIDATE-4+1;
                                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                                count = count-4;
                                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
                }
        }
}
...
*/
// Railgun_Swampshine_BailOut, copyleft 2014-Jan-31, Kaze.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
#define NeedleThreshold2vs4swampLITE 9+10 // Should be bigger than 9. BMH2 works up to this value (inclusive), if bigger then BMH4 takes over. Should be <=255 otherwise the 0¦1 BMH2 should be used.
char * Railgun_Trolldom_1 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        uint32_t ulHashPattern;
        signed long count;

        unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
        unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
        uint32_t i, Gulliver;

        uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
        uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
```

```
       uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

       if (cbPattern > cbTarget) return(NULL);

       if ( cbPattern<4 ) {
             // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
             // ...
             pbTarget = pbTarget+cbPattern;
             ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
             if ( cbPattern==3 ) {
                           for ( ;; ) {
                                         if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                      if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                         }
                                         if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                      pbTarget++;
                                                      if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                         }
                                         pbTarget++;
                                         if (pbTarget > pbTargetMax) return(NULL);
                           }
             } else {
             }
             for ( ;; ) {
                           if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                           if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                           pbTarget++;
                           if (pbTarget > pbTargetMax) return(NULL);
             }
       } else { //if ( cbPattern<4 )
             if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//          }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }
```

```
        AdvanceHopperGrass++;

        pbTarget = pbTarget + AdvanceHopperGrass;
         if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//      } else { //if ( cbPattern<4 )
//          if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
        (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                i=0;
                while (i <= cbTarget-cbPattern) {
                    Gulliver = 1; // 'Gulliver' is the skip
                    //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                    if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                        if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
        == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                                count = cbPattern-4+1;
                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                    count = count-4;
                                if ( count <= 0 ) return(pbTarget+i);
                            }
                        }
                    } else Gulliver = cbPattern-(2-1);
                    i = i + Gulliver;
                    //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
//          } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                    Gulliver = 1; // 'Gulliver' is the skip
                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                                count = cbPattern-4+1;
                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                    count = count-4;
                                if ( count <= 0 ) return(pbTarget+i);
                            }
                        }
                    } else Gulliver = cbPattern-(2-1);
                    i = i + Gulliver;
                    //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
        cbPattern = PRIMALlengthCANDIDATE;
        pbPattern = pbPattern - (PRIMALposition-1);
}
```

```
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
------------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                    ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦---------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                             ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
```

```
|--------------------------------------------------------------------------------|------------------------|
¦ The_Project_Gutenberg_EBook_of_Don        ¦        190 ¦      226 ¦      1654 ¦     1729 ¦      1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                   ¦            ¦          ¦           ¦          ¦           ¦          ¦
¦ 2,347,772 bytes                           ¦            ¦          ¦           ¦          ¦           ¦          ¦
|--------------------------------------------------------------------------------|------------------------|
¦ The_Project_Gutenberg_EBook_of_Dokoe      ¦        582 ¦      760 ¦      3094 ¦     2898 ¦      2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦            ¦          ¦           ¦          ¦           ¦          ¦
¦ 899,425 bytes                             ¦            ¦          ¦           ¦          ¦           ¦          ¦
|--------------------------------------------------------------------------------|------------------------|
¦ Dragonfly_genome_shotgun_sequence         ¦        104 ¦      109 ¦       445 ¦      458 ¦       484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                    ¦            ¦          ¦           ¦          ¦           ¦          ¦
¦ 4,487,433 bytes                           ¦            ¦          ¦           ¦          ¦           ¦          ¦
|--------------------------------------------------------------------------------|------------------------|
¦ LAOTZU_Wu_Wei_(BINARY).pdf                ¦         99 ¦      144 ¦       629 ¦      580 ¦       185 ¦      570 ¦
¦ 954,035 bytes                             ¦            ¦          ¦           ¦          ¦           ¦          ¦
|--------------------------------------------------------------------------------|------------------------|
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00              mov r8d, 65536
//  0012d 44 8b 26        mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                        // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                        // we must switch to 0¦1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
      it is okay to use 'memset'. !
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
                                                            if ( count <= 0 ) return(pbTarget+i);
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
      ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
```

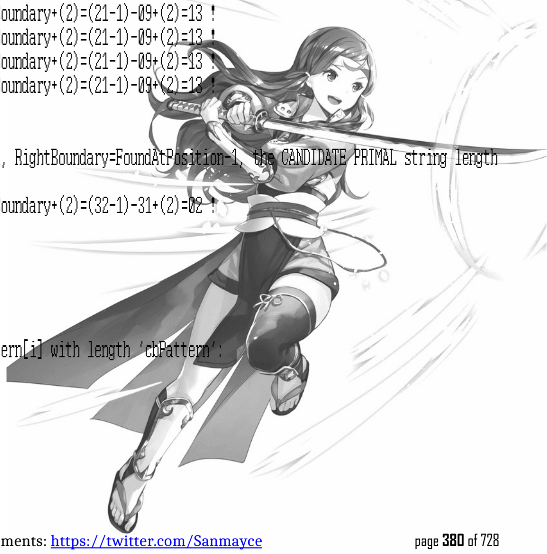
```
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
```

```
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
        pushq   %r15
        .seh_pushreg    %r15
        movl    $65592, %eax
        pushq   %r14
        .seh_pushreg    %r14
        pushq   %r13
        .seh_pushreg    %r13
        pushq   %r12
        .seh_pushreg    %r12
        pushq   %rbp
        .seh_pushreg    %rbp
        pushq   %rdi
        .seh_pushreg    %rdi
        pushq   %rsi
        .seh_pushreg    %rsi
        pushq   %rbx
        .seh_pushreg    %rbx
        call    ___chkstk_ms
        subq    %rax, %rsp
        .seh_stackalloc 65592
        .seh_endprologue
        cmpl    %r9d, %r8d
        movq    %rcx, %rbx
        movq    %rdx, %rdi
        movl    %r8d, %r12d
        movl    %r9d, %esi
        jb      .L118
        movl    (%rdx), %ebp
        leal    -1(%r9), %edx
        movl    $65536, %r8d
        leaq    48(%rsp), %rcx
        movzbl%dl, %edx
        call    memset
        movl    %esi, %r11d
        subl    $1, %r11d
        je      .L119
        xorl    %eax, %eax
        .p2align 4,,10
.L113:
        movzwl(%rdi,%rax), %edx
        movb    %al, 48(%rsp,%rdx)
```

```
	addq	$1, %rax
	cmpl	%eax, %r11d
	ja	.L113
.L112:
	leal	-4(%rsi), %r9d
	movl	%r12d, %r8d
	xorl	%edx, %edx
	leal	-3(%rsi), %eax
	shrl	$2, %r9d
	subl	%esi, %r8d
	leal	-2(%rsi), %r10d
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl(%rbx,%rax), %eax
	movzbl48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
	cmpl	%r8d, %edx
	jbe	.L117
.L118:
	xorl	%eax, %eax
	jmp	.L128
	.p2align 4,,10
.L129:
	movl	%edx, %ecx
	movl	$1, %eax
	leaq	(%rbx,%rcx), %r12
	cmpl	(%r12), %ebp
	jne	.L114
	movl	44(%rsp), %esi
	testl	%esi, %esi
	jle	.L124
	movl	(%r12,%r15), %esi
	cmpl	%esi, (%rdi,%r15)
	jne	.L114
	addq	%r14, %rcx
	xorl	%eax, %eax
	addq	%rbx, %rcx
	jmp	.L116
	.p2align 4,,10
.L132:
	movl	-5(%r13,%rax), %esi
	subq	$4, %rax
	cmpl	-1(%rcx,%rax), %esi
	jne	.L131
.L116:
	cmpq	%rax, %r9
	jne	.L132
.L124:
	movq	%r12, %rax
.L128:
	addq	$65592, %rsp
	popq	%rbx
	popq	%rsi
	popq	%rdi
	popq	%rbp
	popq	%r12
	popq	%r13
	popq	%r14
	popq	%r15
	ret
	.p2align 4,,10
.L131:
	movl	$1, %eax
	jmp	.L114
.L119:
	xorl	%r11d, %r11d
	jmp	.L112
*/
```

```
} //if (cbTarget<777)

            } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv           PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                   PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                 PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                     PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                   PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                   PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                       PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                       PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                 PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                     PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
       FoundAtPosition = cbPattern - ((4)-1) + 1;
       PRIMALpositionCANDIDATE=i;
       while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
             j = PRIMALpositionCANDIDATE + 1;
             while ( j <= (FoundAtPosition-1) ) {
                          if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                          j++;
             }
             PRIMALpositionCANDIDATE++;
       }
       PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
       if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
       if (cbPattern-i+1 <= PRIMALlength) break;
       if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv           PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                   PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                 PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                     PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                   PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                        PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                       PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                     PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                    PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                      000000000111111111122222222223333
                      123456789012345678901234567890123
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
```

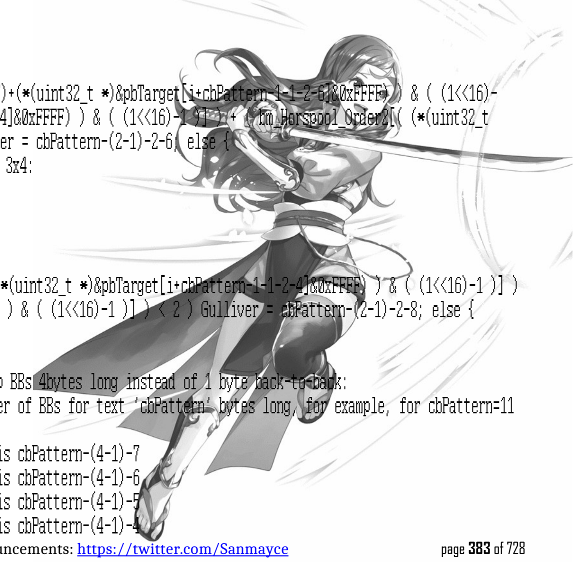
```
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = ‘vvBOOMSHAKALA’


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for ‘12’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for ‘23’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for ‘zx’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for ‘xc’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for ‘23’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for ‘34’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for ‘zx’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for ‘xc’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for ‘xc’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = ‘1234567890qwertyuiopasdfghjklzxcv’



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for ‘vv’, FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = ‘vv’



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for ‘vB’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for ‘BO’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for ‘OO’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for ‘OM’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for ‘MS’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for ‘SH’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for ‘HA’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvvv  ! Searching for ‘AK’, FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvvv  ! Searching for ‘KA’, FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvvv  ! Searching for ‘AL’, FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvvv  ! Searching for ‘LA’, FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for ‘vv’, FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = ‘vvBOOMSHAKALA’
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length ‘PRIMALlength’ and compare the pbPattern[i] with length ‘cbPattern’:
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
```



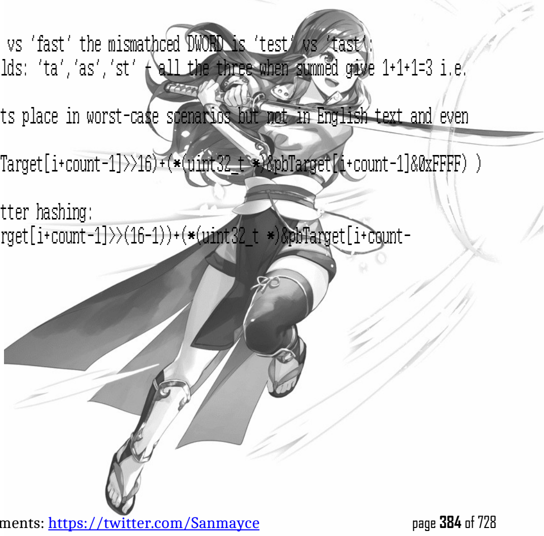
```
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
```

```
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
}
*/
// Revision 2 commented section ]

    if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//       00             mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;

  if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
  } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                        if ( count <= 0 ) return(pbTarget+i);
  }
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);

    } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

        // BMH pseudo-order 4, needle should be >=8+2:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
  example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
```

```
                           //"st f"
                           //"t fo"
                           //" fox"
                           //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                           //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                           // Above line is replaced by next one with better hashing:
                           for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                           i=0;
                           while (i <= cbTarget-cbPattern) {
                                        Gulliver = 1;
                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                                        // Above line is replaced by next one with better hashing:
                                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
       #1
                                                      //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
       0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                                      // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are ‘ab’ and ‘cd’ i.e. [2x][2a][2b][2c][2d] then
       the middle DWORD is ‘bc’.
                                                      // The respective offsets (backwards) are: -10/-8/-6/-4 for ‘xa’/‘ab’/‘bc’/‘cd’.
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
       + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                      // Above line is replaced by next one with better hashing:
                                                      // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
       for ASCII texts with no symbols in range 128-255.
                                                      // Example for genomesque pair TT+TT being shifted by (16-1):
                                                      // T           = 01010100
                                                      // TT          = 01010100 01010100
                                                      // TTTT        = 01010100 01010100 01010100 01010100
                                                      // TTTT>>16    = 00000000 00000000 01010100 01010100
                                                      // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
       0 for English texts & ‘ACGT’ data.
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                      // ‘Maximus’ uses branched ‘if’, again.
                                                      if ( \
                                                      ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
       1 )] ) == 0 \
                                                      || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
       1 )] ) == 0 \
                                                      ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                                      // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                         |
//    DWORD #3                                                                         |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                     |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                         |
//    DWORD #3                                                                         |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                     |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
       + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                                   if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                                // Order 4 [
                                                                   // Let’s try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                                   // Inhere we are using order 4, ‘cbPattern - Order + 1’ is the number of BBs for text ‘cbPattern’ bytes long, for example, for cbPattern=11
       ‘fastest fox’ and Order=4 we have BBs = 11-4+1=8:
                                                                   //0:"fast" if the comparison failed here, ‘count’ is 1; ‘Gulliver’ is cbPattern-(4-1)-7
                                                                   //1:"aste" if the comparison failed here, ‘count’ is 2; ‘Gulliver’ is cbPattern-(4-1)-6
                                                                   //2:"stes" if the comparison failed here, ‘count’ is 3; ‘Gulliver’ is cbPattern-(4-1)-5
                                                                   //3:"test" if the comparison failed here, ‘count’ is 4; ‘Gulliver’ is cbPattern-(4-1)-4
```

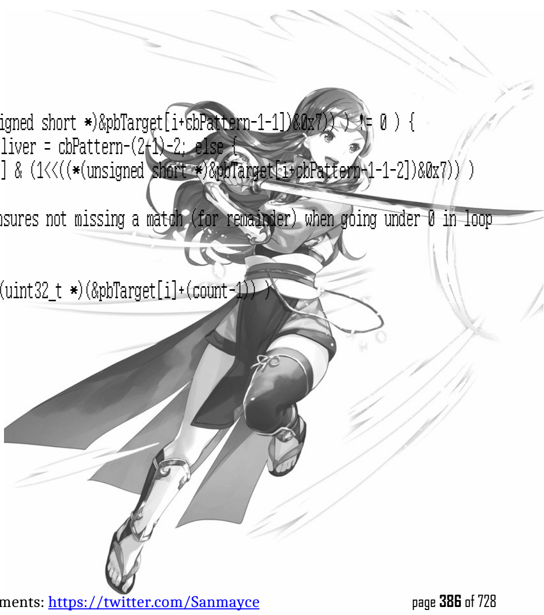


```
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                // Below comparison is UNIdirectional:
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }

                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it

    should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,

    as it should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or

    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.

    Gulliver remains 1.
                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even

    not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )

    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
```



```
                } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } //if ( cbPattern<4 )
}
/*
// For short needles, and mainly haystacks, 'Doublet' is quite effective. Consider it or 'Quadruplet'.
// Fixed version from 2012-Feb-27.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
char * Railgun_Doublet (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        register uint32_t ulHashPattern;
        uint32_t ulHashTarget, count, countSTATIC;

        if (cbPattern > cbTarget) return(NULL);

        countSTATIC = cbPattern-2;

        pbTarget = pbTarget+cbPattern;
        ulHashPattern = (*(uint16_t *)(pbPattern));

        for ( ;; ) {
                if ( ulHashPattern == (*(uint16_t *)(pbTarget-cbPattern)) ) {
                                count = countSTATIC;
                                while ( count && *(char *)(pbPattern+2+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+2+(countSTATIC-count)) ) {
                                                count--;
                                }
                                if ( count == 0 ) return((pbTarget-cbPattern));
                }
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }
}
*/

char * Railgun_Trolldom_2 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        uint32_t ulHashPattern;
        signed long count;

        unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
        unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
        uint32_t i, Gulliver;

        uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
        uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
        uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

        if (cbPattern > cbTarget) return(NULL);

        if ( cbPattern<4 ) {
                // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
                // ...
                pbTarget = pbTarget+cbPattern;
                ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
                if ( cbPattern==3 ) {
                                for ( ;; ) {
                                                if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                                }
                                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                                pbTarget++;
                                                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                                }
                                                pbTarget++;
                                                if (pbTarget > pbTargetMax) return(NULL);
                                }
                } else {
                }
                for ( ;; ) {
                                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                                pbTarget++;
                                if (pbTarget > pbTargetMax) return(NULL);
                }
        } else { //if ( cbPattern<4 )
```

```
            if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//     } else { //if ( cbPattern<4 )
//          if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                        //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
       (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                Gulliver = 1; // 'Gulliver' is the skip
                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
       == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                        count = cbPattern-4+1;
                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                count = count-4;
                                                        if ( count <= 0 ) return(pbTarget+i);
                                                }
                                        }
                                } else Gulliver = cbPattern-(2-1);
                                i = i + Gulliver;
                                //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
//          } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                        // BMH order 2, needle should be >=4:
```



```
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                Gulliver = 1; // 'Gulliver' is the skip
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                        count = cbPattern-4+1;
                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                count = count-4;
                                                        if ( count <= 0 ) return(pbTarget+i);
                                                }
                                        }
                                } else Gulliver = cbPattern-(2-1);
                                i = i + Gulliver;
                                //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
       cbPattern = PRIMALlengthCANDIDATE;
       pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                Gulliver = 1; // 'Gulliver' is the skip
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                        count = cbPattern-4+1;
                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                count = count-4;
                                                        if ( count <= 0 ) return(pbTarget+i);
                                                }
                                        }
                                } else Gulliver = cbPattern-(2-1);
                                i = i + Gulliver;
                                //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                        // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                        // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                        // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
       (Order-1) chars i.e. one char for Order 2
                                if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
```

```
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
       // The line below is BUGGY:
       //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
       // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
       //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
       if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
             if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                           count = PRIMALlengthCANDIDATE-4+1;
                           while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                         count = count-4;
                           if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
             }
       }
}
                                                     }
                                                     Gulliver = 1;
                                        } else
                                                     Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                        }
                                        i = i + Gulliver;
                                        //GlobalI++; // Comment it, it is only for stats.
                           }
                           return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
---------------------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                         ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom       ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                                  ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don        ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                   ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe      ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence         ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                    ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf                ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                           ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                           for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                           // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40         add eax, 64
//  00142 66 0f 7f 44 14
//        60               movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00   cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70               movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00      movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00      movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2            mov edx, eax
//  00167 72 d6            jb .B5.21
                           //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                           // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00               mov r8d, 65536
//  0012d 44 8b 26         mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff   call _intel_fast_memset
                           // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
```

```
                        // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
        it is okay to use 'memset'. !
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
        (Order-1) chars i.e. one char for Order 2
                                if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                        Gulliver = 1;
                                } else
                                        Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                }
                                i = i + Gulliver;
                                //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
        ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
```

```
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
```

```
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
        pushq %r15
        .seh_pushreg        %r15
        movl  $65592, %eax
        pushq %r14
        .seh_pushreg        %r14
        pushq %r13
        .seh_pushreg        %r13
        pushq %r12
        .seh_pushreg        %r12
        pushq %rbp
        .seh_pushreg        %rbp
        pushq %rdi
        .seh_pushreg        %rdi
        pushq %rsi
        .seh_pushreg        %rsi
        pushq %rbx
        .seh_pushreg        %rbx
        call  ___chkstk_ms
        subq  %rax, %rsp
        .seh_stackalloc     65592
        .seh_endprologue
        cmpl  %r9d, %r8d
        movq  %rcx, %rbx
        movq  %rdx, %rdi
        movl  %r8d, %r12d
        movl  %r9d, %esi
        jb    .L118
        movl  (%rdx), %ebp
        leal  -1(%r9), %edx
        movl  $65536, %r8d
        leaq  48(%rsp), %rcx
        movzbl%dl, %edx
        call  memset
        movl  %esi, %r11d
        subl  $1, %r11d
        je    .L119
        xorl  %eax, %eax
        .p2align 4,,10
.L113:
        movzwl(%rdi,%rax), %edx
        movb  %al, 48(%rsp,%rdx)
        addq  $1, %rax
        cmpl  %eax, %r11d
        ja    .L113
.L112:
        leal  -4(%rsi), %r9d
        movl  %r12d, %r8d
        xorl  %edx, %edx
        leal  -3(%rsi), %eax
        shrl  $2, %r9d
        subl  %esi, %r8d
        leal  -2(%rsi), %r10d
        movslq%eax, %r14
        negq  %r9
        movl  %eax, 44(%rsp)
        leaq  -1(%r14), %r15
        salq  $2, %r9
        leaq  (%rdi,%r14), %r13
        jmp   .L117
        .p2align 4,,10
.L130:
        movl  %r10d, %eax
        subl  %ecx, %eax
        cmpl  %r10d, %ecx
        je    .L129
.L114:
        addl  %eax, %edx
        cmpl  %r8d, %edx
        ja    .L118
.L117:
        leal  (%rdx,%r10), %eax
        movzwl(%rbx,%rax), %eax
        movzbl48(%rsp,%rax), %ecx
        cmpl  %r11d, %ecx
        jne   .L130
        movl  %r11d, %eax
        addl  %eax, %edx
        cmpl  %r8d, %edx
        jbe   .L117
.L118:
        xorl  %eax, %eax
        jmp   .L128
        .p2align 4,,10
.L129:
```

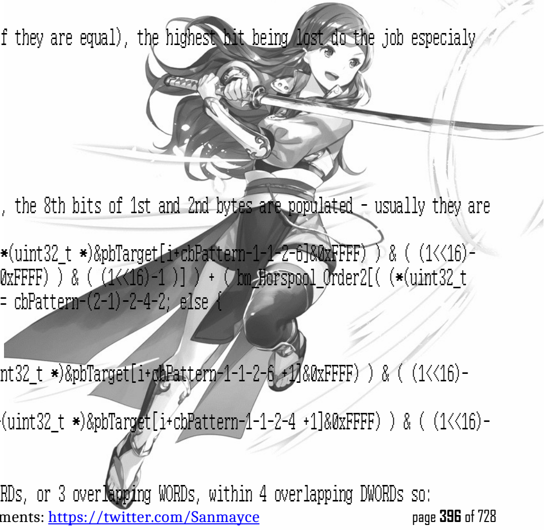
```
        movl  %edx, %ecx
        movl  $1, %eax
        leaq  (%rbx,%rcx), %r12
        cmpl  (%r12), %ebp
        jne   .L114
        movl  44(%rsp), %esi
        testl %esi, %esi
        jle   .L124
        movl  (%r12,%r15), %esi
        cmpl  %esi, (%rdi,%r15)
        jne   .L114
        addq  %r14, %rcx
        xorl  %eax, %eax
        addq  %rbx, %rcx
        jmp   .L116
        .p2align 4,,10
.L132:
        movl  -5(%r13,%rax), %esi
        subq  $4, %rax
        cmpl  -1(%rcx,%rax), %esi
        jne   .L131
.L116:
        cmpq  %rax, %r9
        jne   .L132
.L124:
        movq  %r12, %rax
.L128:
        addq  $65592, %rsp
        popq  %rbx
        popq  %rsi
        popq  %rdi
        popq  %rbp
        popq  %r12
        popq  %r13
        popq  %r14
        popq  %r15
        ret
        .p2align 4,,10
.L131:
        movl  $1, %eax
        jmp   .L114
.L119:
        xorl  %r11d, %r11d
        jmp   .L112
*/

} //if (cbTarget<777)

                } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                   PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                 PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                     PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                   PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                   PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                       PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                       PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                 PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                     PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
        while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
                j = PRIMALpositionCANDIDATE + 1;
                while ( j <= (FoundAtPosition-1) ) {
                        if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                        j++;
                }
                PRIMALpositionCANDIDATE++;
        }
        PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
        if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
        if (cbPattern-i+1 <= PRIMALlength) break;
        if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
```

```
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv           PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                   PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                 PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                     PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                   PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                        PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                       PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                     PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                    PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       000000000111111111122222222223333
                       123456789012345678901234567890123
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
```

```
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                           ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                           for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                           for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                           i=0;
                           while (i <= cbTarget-cbPattern) {
                                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
       (Order-1) chars i.e. one char for Order 2
                                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                     if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                  count = cbPattern-4+1;
                                                                  while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                               count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
       // The line below is BUGGY:
       //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
       // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
       //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
       if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
             if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                           count = PRIMALlengthCANDIDATE-4+1;
                           while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                        count = count-4;
                           if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
             }
       }
}
                                                     }
                                                     Gulliver = 1;
```

```
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00            mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
```

```
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
        // The line below is BUGGY:
        //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
        // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
        //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
        if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
            if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                        count = PRIMALlengthCANDIDATE-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                count = count-4;
                        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
            }
        }
}
        } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                        if ( count <= 0 ) return(pbTarget+i);
        }
                                                }
                                        }
                                } else Gulliver = cbPattern-(2-1);
                                i = i + Gulliver;
                                //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);

            } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                        // BMH pseudo-order 4, needle should be >=8+2:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
        example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                        //"fast"
                        //"aste"
                        //"stes"
                        //"test"
                        //"est "
                        //"st f"
                        //"t fo"
                        //" fox"
                        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                        // Above line is replaced by next one with better hashing:
                        for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                Gulliver = 1;
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                                // Above line is replaced by next one with better hashing:
                                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
        #1
                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
        0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                        // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
        the middle DWORD is 'bc'.
                                        // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
        + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
        *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                        // Above line is replaced by next one with better hashing:
                                        // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
        for ASCII texts with no symbols in range 128-255.
                                        // Example for genomesque pair TT+TT being shifted by (16-1):
                                        // T           = 01010100
                                        // TT          = 01010100 01010100
                                        // TTTT        = 01010100 01010100 01010100 01010100
                                        // TTTT>>16    = 00000000 00000000 01010100 01010100
                                        // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
        0 for English texts & 'ACGT' data.
                                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
        1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
        *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                        // 'Maximus' uses branched 'if', again.
                                        if ( \
                                        ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
        1 )] ) == 0 \
                                        || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
        1 )] ) == 0 \
                                        ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                        // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
```



```
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        ¦
//     DWORD #3                                                                       ¦
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!   ¦<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  ¦
//         DWORD #2                                                                      ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!      ¦<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  ¦
//             DWORD #1                                                                     ¦
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = --------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        ¦
//     DWORD #3                                                                       ¦
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!   ¦<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  ¦
//         DWORD #2                                                                      ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      ¦<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  ¦
//             DWORD #1                                                                     ¦
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = --------------------------------
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
       + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                                 // Order 4 [
                                                                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
       'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                                                 count = cbPattern-4+1;
                                                                                 // Below comparison is UNIdirectional:
                                                                                 while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                               count = count-4;

       if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
       // The line below is BUGGY:
       //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
       // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
       //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
       if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
              if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                           count = PRIMALlengthCANDIDATE-4+1;
                           while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                         count = count-4;
                           if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
              }
       }
```

```
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }


                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,

    as it should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or

    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
    Gulliver remains 1.

                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even

    not in ACGT data, that's why I commented it in original 'Shockeroo'.

                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )

    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should

                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_3 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;

    uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
    uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
    uint32_t j, FoundAtPosition;

// Quadruplet [
  //char * pbTargetMax = pbTarget + cbTarget;
  //register unsigned long  ulHashPattern;
  uint32_t ulHashTarget;
  //unsigned long count;
  uint32_t countSTATIC;
  unsigned char SINGLET;
  uint32_t Quadruplet2nd;
  uint32_t Quadruplet3rd;
  uint32_t Quadruplet4th;
  uint32_t AdvanceHopperGrass;
// Quadruplet ]

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {
        // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
        // ...
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
```

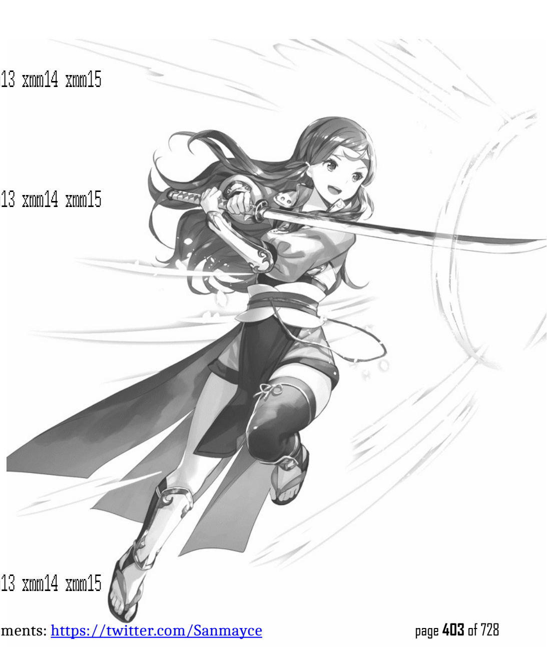
```
                                    if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                pbTarget++;
                                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                    }
                                    pbTarget++;
                                    if (pbTarget > pbTargetMax) return(NULL);
                        }
            } else {
            }
            for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
            }
      } else { //if ( cbPattern<4 )
            if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
       pbTarget = pbTarget+cbPattern;
       ulHashPattern = *(uint32_t *)(pbPattern);
//       countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
      AdvanceHopperGrass = 0;
      ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

       if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//        count = countSTATIC;
//        while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//           if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//             count--;
//        }
        count = cbPattern-1;
        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
             if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
              count--;
        }
        if ( count == 0) return((pbTarget-cbPattern));
       } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
        AdvanceHopperGrass++;
        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
            AdvanceHopperGrass++;
            if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
        }
    }
      }

      AdvanceHopperGrass++;

      pbTarget = pbTarget + AdvanceHopperGrass;
       if (pbTarget > pbTargetMax)
          return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//    } else { //if ( cbPattern<4 )
//          if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                        //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
      (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                    if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
      == 0 ) Gulliver = cbPattern-(2-1)-2; else {
```

```
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
                        if ( count <= 0 ) return(pbTarget+i);
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
//  } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
                        if ( count <= 0 ) return(pbTarget+i);
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
    cbPattern = PRIMALlengthCANDIDATE;
    pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
                        if ( count <= 0 ) return(pbTarget+i);
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
```

```
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
--------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                      ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
|--------------------------------------------------------------------------------|-----------------------|
¦ Compiler                               ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
|--------------------------------------------------------------------------------|-----------------------|
¦ The_Project_Gutenberg_EBook_of_Don     ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                        ¦            ¦          ¦            ¦          ¦            ¦          ¦
|--------------------------------------------------------------------------------|-----------------------|
¦ The_Project_Gutenberg_EBook_of_Dokoe   ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦         ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                          ¦            ¦          ¦            ¦          ¦            ¦          ¦
|--------------------------------------------------------------------------------|-----------------------|
¦ Dragonfly_genome_shotgun_sequence      ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                 ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                        ¦            ¦          ¦            ¦          ¦            ¦          ¦
|--------------------------------------------------------------------------------|-----------------------|
¦ LAOTZU_Wu_Wei_(BINARY).pdf             ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                          ¦            ¦          ¦            ¦          ¦            ¦          ¦
|--------------------------------------------------------------------------------|-----------------------|
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
```

```
//  00148 3d 00 00 01 00   cmp eax, 65536
//  0014d 66 0f 7f 44 14
//       70                movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//       80 00 00 00       movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//       90 00 00 00       movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2            mov edx, eax
//  00167 72 d6            jb .B5.21
                           //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                           // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//       00                mov r8d, 65536
//  0012d 44 8b 26         mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff   call _intel_fast_memset
                           // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                           // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
      it is okay to use 'memset'. !
                           for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                           i=0;
                           while (i <= cbTarget-cbPattern) {
                                       Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                       if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                                       if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                   if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                               count = cbPattern-4+1;
                                                               while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                           count = count-4;
                                                               if ( count <= 0 ) return(pbTarget+i);
                                                   }
                                                   Gulliver = 1;
                                       } else
                                                   Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                       }
                                       i = i + Gulliver;
                                       //GlobalI++; // Comment it, it is only for stats.
                           }
                           return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
```

```
        test    r15d, r15d                              ;3515.4
        jbe     .B6.10        ; Prob 15%                ;3515.4
                              ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                       ; Preds .B6.6 .B6.8
        lea     eax, DWORD PTR [r10+r10]                ;3515.36
        movzx   ebx, WORD PTR [rax+rdx]                 ;3515.75
        mov     BYTE PTR [48+rsp+rbx], al               ;3515.36
        lea     eax, DWORD PTR [1+r10+r10]              ;3515.36
        inc     r10d                                    ;3515.4
        cmp     r10d, r15d                              ;3515.4
        movzx   ebx, WORD PTR [rax+rdx]                 ;3515.75
        mov     BYTE PTR [48+rsp+rbx], al               ;3515.36
        jb      .B6.8         ; Prob 64%                ;3515.4
                              ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                       ; Preds .B6.8
        lea     eax, DWORD PTR [1+r10+r10]              ;3515.4
                              ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                      ; Preds .B6.9 .B6.6
        dec     eax                                     ;3515.36
        cmp     eax, r11d                               ;3515.4
        jae     .B6.12        ; Prob 15%                ;3515.4
                              ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                      ; Preds .B6.10
        movzx   r10d, WORD PTR [rax+rdx]                ;3515.75
        mov     BYTE PTR [48+rsp+r10], al               ;3515.36
                              ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                      ; Preds .B6.5 .B6.10 .B6.11
        xor     r10d, r10d                              ;3516.4
        lea     r15d, DWORD PTR [-3+r9]                 ;3522.27
        movsxd  r15, r15d                               ;3522.7
        sub     r8d, r9d                                ;3517.16
        lea     r11d, DWORD PTR [-2+r9]                 ;3520.32
                              ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                      ; Preds .B6.12 .B6.24
        lea     eax, DWORD PTR [-2+r9+r10]              ;3518.78
        movzx   ebx, WORD PTR [rax+rcx]                 ;3518.55
        movzx   eax, BYTE PTR [48+rsp+rbx]              ;3518.16
        cmp     eax, ebp                                ;3519.32
        je      .B6.24        ; Prob 50%                ;3519.32
                              ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                      ; Preds .B6.13
        cmp     eax, r11d                               ;3520.32
        jne     .B6.23        ; Prob 62%                ;3520.32
                              ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                      ; Preds .B6.14
        mov     eax, r10d                               ;3521.25
        add     rax, rcx                                ;3521.25
        cmp     r13d, DWORD PTR [rax]                   ;3521.40
        je      .B6.17        ; Prob 50%                ;3521.40
                              ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                      ; Preds .B6.26 .B6.15
        mov     eax, 1                                  ;3527.6
        jmp     .B6.24        ; Prob 100%               ;3527.6
                              ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                      ; Preds .B6.15
        mov     rbx, r15                                ;3522.7
        test    r15, r15                                ;3523.23
        jle     .B6.22        ; Prob 2%                 ;3523.23
                              ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                      ; Preds .B6.17
        mov     QWORD PTR [32+rsp], rsi                 ;
                              ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                      ; Preds .B6.20 .B6.18
        mov     esi, DWORD PTR [-1+rbx+rdx]             ;3523.58
        cmp     esi, DWORD PTR [-1+rbx+rax]             ;3523.79
        jne     .B6.26        ; Prob 20%                ;3523.79
                              ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                      ; Preds .B6.19
        add     rbx, -4                                 ;3524.22
        test    rbx, rbx                                ;3523.23
        jg      .B6.19        ; Prob 82%                ;3523.23
                              ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                      ; Preds .B6.20
        mov     rsi, QWORD PTR [32+rsp]                 ;
                              ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                      ; Preds .B6.17 .B6.21
        add     rsp, 65592                              ;3525.32
        pop     rbp                                     ;3525.32
        pop     r15                                     ;3525.32
        pop     r13                                     ;3525.32
        pop     rbx                                     ;3525.32
        ret                                             ;3525.32
                              ; LOE
.B6.23::                      ; Preds .B6.14
        neg     eax                                     ;3529.17
        add     eax, r9d                                ;3529.17
        add     eax, -2                                 ;3529.40
                              ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                      ; Preds .B6.16 .B6.23 .B6.13
```



```
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
       pushq %r15
       .seh_pushreg      %r15
       movl  $65592, %eax
       pushq %r14
       .seh_pushreg      %r14
       pushq %r13
       .seh_pushreg      %r13
       pushq %r12
       .seh_pushreg      %r12
       pushq %rbp
       .seh_pushreg      %rbp
       pushq %rdi
       .seh_pushreg      %rdi
       pushq %rsi
       .seh_pushreg      %rsi
       pushq %rbx
       .seh_pushreg      %rbx
       call  ___chkstk_ms
       subq  %rax, %rsp
       .seh_stackalloc   65592
       .seh_endprologue
       cmpl  %r9d, %r8d
       movq  %rcx, %rbx
       movq  %rdx, %rdi
       movl  %r8d, %r12d
       movl  %r9d, %esi
       jb    .L118
       movl  (%rdx), %ebp
       leal  -1(%r9), %edx
       movl  $65536, %r8d
       leaq  48(%rsp), %rcx
       movzbl%dl, %edx
       call  memset
       movl  %esi, %r11d
       subl  $1, %r11d
       je    .L119
       xorl  %eax, %eax
       .p2align 4,,10
.L113:
       movzwl(%rdi,%rax), %edx
       movb  %al, 48(%rsp,%rdx)
       addq  $1, %rax
       cmpl  %eax, %r11d
       ja    .L113
.L112:
       leal  -4(%rsi), %r9d
       movl  %r12d, %r8d
       xorl  %edx, %edx
       leal  -3(%rsi), %eax
       shrl  $2, %r9d
       subl  %esi, %r8d
       leal  -2(%rsi), %r10d
       movslq%eax, %r14
       negq  %r9
       movl  %eax, 44(%rsp)
       leaq  -1(%r14), %r15
       salq  $2, %r9
       leaq  (%rdi,%r14), %r13
       jmp   .L117
       .p2align 4,,10
.L130:
       movl  %r10d, %eax
       subl  %ecx, %eax
       cmpl  %r10d, %ecx
       je    .L129
.L114:
       addl  %eax, %edx
       cmpl  %r8d, %edx
```

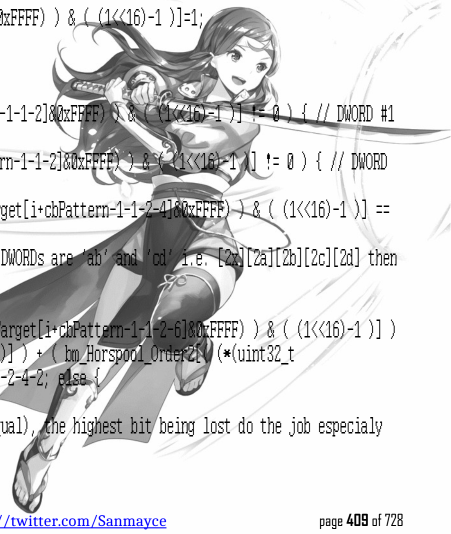
```
        ja      .L118
.L117:
        leal    (%rdx,%r10), %eax
        movzwl  (%rbx,%rax), %eax
        movzbl  48(%rsp,%rax), %ecx
        cmpl    %r11d, %ecx
        jne     .L130
        movl    %r11d, %eax
        addl    %eax, %edx
        cmpl    %r8d, %edx
        jbe     .L117
.L118:
        xorl    %eax, %eax
        jmp     .L128
        .p2align 4,,10
.L129:
        movl    %edx, %ecx
        movl    $1, %eax
        leaq    (%rbx,%rcx), %r12
        cmpl    (%r12), %ebp
        jne     .L114
        movl    44(%rsp), %esi
        testl   %esi, %esi
        jle     .L124
        movl    (%r12,%r15), %esi
        cmpl    %esi, (%rdi,%r15)
        jne     .L114
        addq    %r14, %rcx
        xorl    %eax, %eax
        addq    %rbx, %rcx
        jmp     .L116
        .p2align 4,,10
.L132:
        movl    -5(%r13,%rax), %esi
        subq    $4, %rax
        cmpl    -1(%rcx,%rax), %esi
        jne     .L131
.L116:
        cmpq    %rax, %r9
        jne     .L132
.L124:
        movq    %r12, %rax
.L128:
        addq    $65592, %rsp
        popq    %rbx
        popq    %rsi
        popq    %rdi
        popq    %rbp
        popq    %r12
        popq    %r13
        popq    %r14
        popq    %r15
        ret
        .p2align 4,,10
.L131:
        movl    $1, %eax
        jmp     .L114
.L119:
        xorl    %r11d, %r11d
        jmp     .L112
*/

} //if (cbTarget<777)

                } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
```

```
       while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
             j = PRIMALpositionCANDIDATE + 1;
             while ( j <= (FoundAtPosition-1) ) {
                           if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                           j++;
             }
             PRIMALpositionCANDIDATE++;
       }
       PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
       if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
       if (cbPattern-i+1 <= PRIMALlength) break;
       if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                    PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                  PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                      PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                    PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                         PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                        PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                      PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                      PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                     PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       00000000011111111112222222222333
                       12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
```

```
    PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
    Result:
    PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'


    PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
    Result:
    PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                    ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                    for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                    for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                    i=0;
                    while (i <= cbTarget-cbPattern) {
                              Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                              if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                              if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                  count = cbPattern-4+1;
                                                  while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                            count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
      // The line below is BUGGY:
```

```
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00            mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
```

```
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
    #1
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
    the middle DWORD is 'bc'.
                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // Above line is replaced by next one with better hashing:
                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
    for ASCII texts with no symbols in range 128-255.
                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                // T            = 01010100
```



```
                                                     // TT          = 01010100 01010100
                                                     // TTTT        = 01010100 01010100 01010100 01010100
                                                     // TTTT>>16    = 00000000 00000000 01010100 01010100
                                                     // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
       0 for English texts & 'ACGT' data.
                                                     //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                     // 'Maximus' uses branched 'if', again.
                                                     if ( \
                                                     ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
       1 )] ) == 0 \

                                                     || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
       1 )] ) == 0 \

                                                     ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                                     // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                      ¦
//    DWORD #3                                                                      ¦
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!  ¦<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ----------------------  ¦
//        DWORD #2                                                                     ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!     ¦<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = -------------------------  ¦
//            DWORD #1                                                                    ¦
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                               ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ----------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                      ¦
//    DWORD #3                                                                      ¦
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!  ¦<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ----------------------  ¦
//        DWORD #2                                                                     ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!     ¦<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = -------------------------  ¦
//            DWORD #1                                                                    ¦
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                               ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ----------------------------
                                                     //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//        DWORD #2
//                DWORD #1
                                                     //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
       + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                                  if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                               // Order 4 [
                                                                  // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                                  // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
       'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                                  //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                                  //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                                  //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                                  //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                                  //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                                  //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                                  //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                                  //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                                               count = cbPattern-4+1;
                                                                               // Below comparison is UNIdirectional:
                                                                               while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                            count = count-4;

       if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
```

```
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }


                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
    as it should
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
    Gulliver remains 1.
                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
    not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_4 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;

    uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
    uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
    uint32_t j, FoundAtPosition;

// Quadruplet [
  //char * pbTargetMax = pbTarget + cbTarget;
  //register unsigned long  ulHashPattern;
  uint32_t ulHashTarget;
  //unsigned long count;
  uint32_t countSTATIC;
  unsigned char SINGLET;
  uint32_t Quadruplet2nd;
  uint32_t Quadruplet3rd;
```

```
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

        if (cbPattern > cbTarget) return(NULL);

        if ( cbPattern<4 ) {
              // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
              // ...
              pbTarget = pbTarget+cbPattern;
              ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
              if ( cbPattern==3 ) {
                          for ( ;; ) {
                                      if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                  if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                      }
                                      if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                  pbTarget++;
                                                  if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                      }
                                      pbTarget++;
                                      if (pbTarget > pbTargetMax) return(NULL);
                          }
              } else {
              }
              for ( ;; ) {
                          if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                          if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                          pbTarget++;
                          if (pbTarget > pbTargetMax) return(NULL);
              }
        } else { //if ( cbPattern<4 )
              if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//     } else { //if ( cbPattern<4 )
```

```
//      if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
            // BMH order 2, needle should be >=4:
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
            for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
            //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
            for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
  (1<<((*(unsigned short *)(pbPattern+i))&0x7));
            i=0;
            while (i <= cbTarget-cbPattern) {
                Gulliver = 1; // 'Gulliver' is the skip
                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                    //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
  == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                            count = cbPattern-4+1;
                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                count = count-4;
                            if ( count <= 0 ) return(pbTarget+i);
                        }
                    }
                } else Gulliver = cbPattern-(2-1);
                i = i + Gulliver;
                //GlobalI++; // Comment it, it is only for stats.
            }
            return(NULL);
//      } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
            // BMH order 2, needle should be >=4:
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//          for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
            for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
            i=0;
            while (i <= cbTarget-cbPattern) {
                Gulliver = 1; // 'Gulliver' is the skip
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                            count = cbPattern-4+1;
                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                count = count-4;
                            if ( count <= 0 ) return(pbTarget+i);
                        }
                    }
                } else Gulliver = cbPattern-(2-1);
                i = i + Gulliver;
                //GlobalI++; // Comment it, it is only for stats.
            }
            return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
    cbPattern = PRIMALlengthCANDIDATE;
    pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
            // BMH order 2, needle should be >=4:
            ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
            for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
            for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
            i=0;
            while (i <= cbTarget-cbPattern) {
                Gulliver = 1; // 'Gulliver' is the skip
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
```

```
  below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
                        if ( count <= 0 ) return(pbTarget+i);
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
        // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
        // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
        // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
            if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
  (Order-1) chars i.e. one char for Order 2
            if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                    count = cbPattern-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
--------------------------------------------------------------------------------------------------
¦ testfile\Searcher                     ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦---------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                              ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦---------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don    ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt               ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                       ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe  ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦        ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence     ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                ¦            ¦          ¦            ¦          ¦            ¦          ¦
```

```
¦ 4,487,433 bytes                          ¦           ¦          ¦            ¦          ¦            ¦          ¦
|--------------------------------------------------------------------------------------------|-----------------------|
¦ LAOTZU_Wu_Wei_(BINARY).pdf               ¦        99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                            ¦           ¦          ¦            ¦          ¦            ¦          ¦
|--------------------------------------------------------------------------------------------|-----------------------|
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00              mov r8d, 65536
//  0012d 44 8b 26        mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                        // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                        // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
      it is okay to use 'memset'. !
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
                                                            if ( count <= 0 ) return(pbTarget+i);
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
```

```
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
```

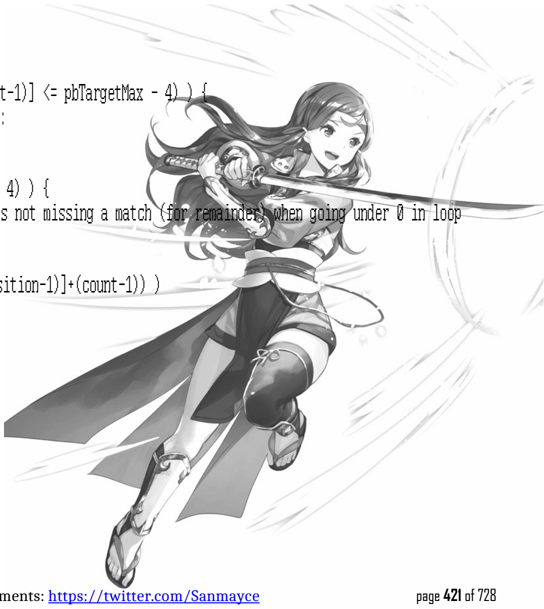
```
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
       pushq %r15
       .seh_pushreg       %r15
       movl  $65592, %eax
       pushq %r14
       .seh_pushreg       %r14
       pushq %r13
       .seh_pushreg       %r13
       pushq %r12
       .seh_pushreg       %r12
       pushq %rbp
       .seh_pushreg       %rbp
       pushq %rdi
       .seh_pushreg       %rdi
       pushq %rsi
       .seh_pushreg       %rsi
       pushq %rbx
       .seh_pushreg       %rbx
       call  ___chkstk_ms
       subq  %rax, %rsp
       .seh_stackalloc    65592
       .seh_endprologue
       cmpl  %r9d, %r8d
       movq  %rcx, %rbx
       movq  %rdx, %rdi
       movl  %r8d, %r12d
       movl  %r9d, %esi
       jb    .L118
       movl  (%rdx), %ebp
       leal  -1(%r9), %edx
       movl  $65536, %r8d
       leaq  48(%rsp), %rcx
       movzbl%dl, %edx
       call  memset
       movl  %esi, %r11d
       subl  $1, %r11d
       je    .L119
       xorl  %eax, %eax
       .p2align 4,,10
.L113:
       movzwl(%rdi,%rax), %edx
       movb  %al, 48(%rsp,%rdx)
       addq  $1, %rax
       cmpl  %eax, %r11d
       ja    .L113
.L112:
       leal  -4(%rsi), %r9d
       movl  %r12d, %r8d
       xorl  %edx, %edx
       leal  -3(%rsi), %eax
       shrl  $2, %r9d
       subl  %esi, %r8d
       leal  -2(%rsi), %r10d
```

```
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl(%rbx,%rax), %eax
	movzbl48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
	cmpl	%r8d, %edx
	jbe	.L117
.L118:
	xorl	%eax, %eax
	jmp	.L128
	.p2align 4,,10
.L129:
	movl	%edx, %ecx
	movl	$1, %eax
	leaq	(%rbx,%rcx), %r12
	cmpl	(%r12), %ebp
	jne	.L114
	movl	44(%rsp), %esi
	testl %esi, %esi
	jle	.L124
	movl	(%r12,%r15), %esi
	cmpl	%esi, (%rdi,%r15)
	jne	.L114
	addq	%r14, %rcx
	xorl	%eax, %eax
	addq	%rbx, %rcx
	jmp	.L116
	.p2align 4,,10
.L132:
	movl	-5(%r13,%rax), %esi
	subq	$4, %rax
	cmpl	-1(%rcx,%rax), %esi
	jne	.L131
.L116:
	cmpq	%rax, %r9
	jne	.L132
.L124:
	movq	%r12, %rax
.L128:
	addq	$65592, %rsp
	popq	%rbx
	popq	%rsi
	popq	%rdi
	popq	%rbp
	popq	%r12
	popq	%r13
	popq	%r14
	popq	%r15
	ret
	.p2align 4,,10
.L131:
	movl	$1, %eax
	jmp	.L114
.L119:
	xorl	%r11d, %r11d
	jmp	.L112
*/

} //if (cbTarget<777)

		} else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
```

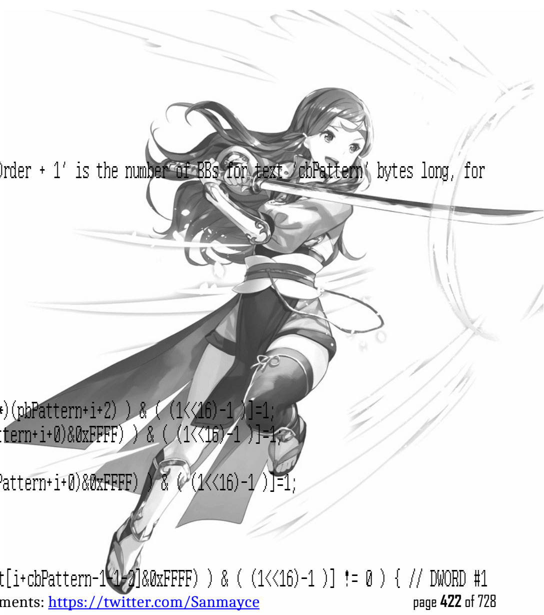
```
// Needle: alfalfa                                     PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                   PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                   PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                       PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                       PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                 PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                     PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
       FoundAtPosition = cbPattern - ((4)-1) + 1;
       PRIMALpositionCANDIDATE=i;
       while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
             j = PRIMALpositionCANDIDATE + 1;
             while ( j <= (FoundAtPosition-1) ) {
                          if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                          j++;
             }
             PRIMALpositionCANDIDATE++;
       }
       PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
       if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
       if (cbPattern-i+1 <= PRIMALlength) break;
       if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv           PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                   PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                 PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                     PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                   PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                        PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                       PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                     PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                    PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       00000000011111111112222222222333
                       12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
```

```
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern';
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
       (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
```

```
                    count = cbPattern-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
} else {
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
```



```
}
*/
// Revision 2 commented section ]

    if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00            mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;

  if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
  } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                        if ( count <= 0 ) return(pbTarget+i);
  }
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);

    } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

        // BMH pseudo-order 4, needle should be >=8+2:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
  example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
        //"st f"
        //"t fo"
        //" fox"
        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        // Above line is replaced by next one with better hashing:
        for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
```



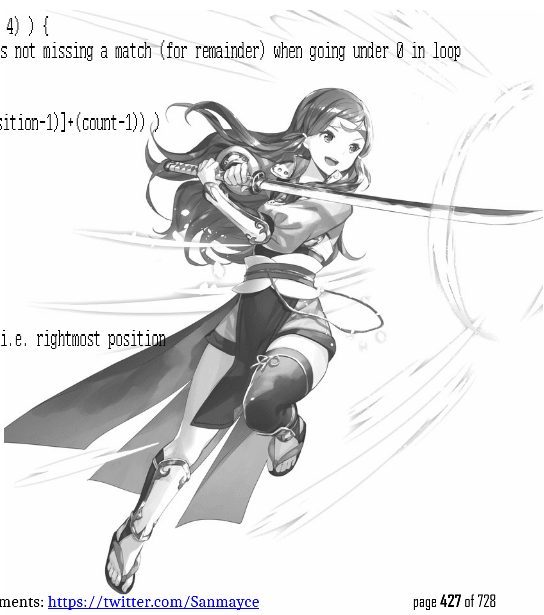
```
                                        // Above line is replaced by next one with better hashing:
                                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
       #1
                                                      //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
       0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                                      // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
       the middle DWORD is 'bc'.
                                                      // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
       + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                      // Above line is replaced by next one with better hashing:
                                                      // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
       for ASCII texts with no symbols in range 128-255.
                                                      // Example for genomesque pair TT+TT being shifted by (16-1):
                                                      // T           = 01010100
                                                      // TT          = 01010100 01010100
                                                      // TTTT        = 01010100 01010100 01010100 01010100
                                                      // TTTT>>16    = 00000000 00000000 01010100 01010100
                                                      // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
       0 for English texts & 'ACGT' data.
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                      // 'Maximus' uses branched 'if', again.
                                                      if ( \
                                                      ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
       1 )] ) == 0 \
                                                      || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
       1 )] ) == 0 \
                                                      ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                                      // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                      ¦
//    DWORD #3                                                                       ¦
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!  ¦<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ---------------------  ¦
//        DWORD #2                                                                     ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!    ¦<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ------------------------  ¦
//            DWORD #1                                                                    ¦
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                              ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ---------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                      ¦
//    DWORD #3                                                                       ¦
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!  ¦<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ---------------------  ¦
//        DWORD #2                                                                     ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!    ¦<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ------------------------  ¦
//            DWORD #1                                                                    ¦
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                              ¦
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ---------------------------
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//        DWORD #2
//                DWORD #1
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
       + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                                   if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                                // Order 4 [
                                                                   // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                                   // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
       'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                                   //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                                   //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                                   //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                                   //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                                   //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                                   //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                                   //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                                   //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                                                count = cbPattern-4+1;
                                                                                // Below comparison is UNIdirectional:
                                                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                             count = count-4;

       if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
```

```
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }


                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it

    should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,

    as it should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or

    bigger, as it should

                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.

    Gulliver remains 1.

                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even

    not in ACGT data, that's why I commented it in original 'Shockeroo'.

                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )

    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should

                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_5 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
```

```
       unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
       uint32_t i, Gulliver;

       uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
       uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
       uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

       if (cbPattern > cbTarget) return(NULL);

       if ( cbPattern<4 ) {
             // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
             // ...
             pbTarget = pbTarget+cbPattern;
             ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
             if ( cbPattern==3 ) {
                           for ( ;; ) {
                                         if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                      if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                         }
                                         if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                      pbTarget++;
                                                      if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                         }
                                         pbTarget++;
                                         if (pbTarget > pbTargetMax) return(NULL);
                           }
             } else {
             }
             for ( ;; ) {
                           if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                           if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                           pbTarget++;
                           if (pbTarget > pbTargetMax) return(NULL);
             }
       } else { //if ( cbPattern<4 )
             if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//              if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//          }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
               if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
                count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
```

```
          AdvanceHopperGrass++;
          if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
       }
   }
     }

     AdvanceHopperGrass++;

     pbTarget = pbTarget + AdvanceHopperGrass;
      if (pbTarget > pbTargetMax)
         return(NULL);
   }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//   } else { //if ( cbPattern<4 )
//        if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                    // BMH order 2, needle should be >=4:
                    ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                    //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                    for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                    //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                    for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
     (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                    i=0;
                    while (i <= cbTarget-cbPattern) {
                              Gulliver = 1; // 'Gulliver' is the skip
                              //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                              if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                       //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                       if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
     == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                 if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
     below:
                                                           count = cbPattern-4+1;
                                                           while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                     count = count-4;
                                                           if ( count <= 0 ) return(pbTarget+i);
                                                 }
                                       }
                              } else Gulliver = cbPattern-(2-1);
                              i = i + Gulliver;
                              //GlobalI++; // Comment it, it is only for stats.
                    }
                    return(NULL);
//        } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                    // BMH order 2, needle should be >=4:
                    ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                  for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                    for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                    i=0;
                    while (i <= cbTarget-cbPattern) {
                              Gulliver = 1; // 'Gulliver' is the skip
                              if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                       if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                 if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
     below:
                                                           count = cbPattern-4+1;
                                                           while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                     count = count-4;
                                                           if ( count <= 0 ) return(pbTarget+i);
                                                 }
                                       }
                              } else Gulliver = cbPattern-(2-1);
                              i = i + Gulliver;
                              //GlobalI++; // Comment it, it is only for stats.
                    }
                    return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
   FoundAtPosition = cbPattern;
   PRIMALpositionCANDIDATE=i;
   while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
      j = PRIMALpositionCANDIDATE + 1;
      while ( j <= (FoundAtPosition-1) ) {
         if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
         j++;
      }
      PRIMALpositionCANDIDATE++;
   }
   PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
   if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
```

```
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
    cbPattern = PRIMALlengthCANDIDATE;
    pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
```



```
So the result on Core 2 Q9550s @2.83GHz:
-------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                     ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                              ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don    ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt               ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                       ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe  ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦        ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence     ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                       ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf            ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40          add eax, 64
//  00142 66 0f 7f 44 14
//        60                movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00    cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70                movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00       movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00       movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2             mov edx, eax
//  00167 72 d6             jb .B5.21
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00                mov r8d, 65536
//  0012d 44 8b 26          mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff    call _intel_fast_memset
                        // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                        // we must switch to 0¦1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
      it is okay to use 'memset'. !
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
                                                            if ( count <= 0 ) return(pbTarget+i);
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
      ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
```

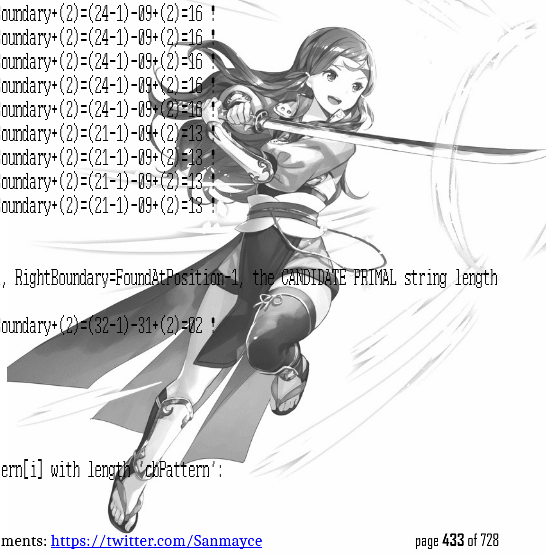
```
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
```

```
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
	pushq	%r15
	.seh_pushreg	%r15
	movl	$65592, %eax
	pushq	%r14
	.seh_pushreg	%r14
	pushq	%r13
	.seh_pushreg	%r13
	pushq	%r12
	.seh_pushreg	%r12
	pushq	%rbp
	.seh_pushreg	%rbp
	pushq	%rdi
	.seh_pushreg	%rdi
	pushq	%rsi
	.seh_pushreg	%rsi
	pushq	%rbx
	.seh_pushreg	%rbx
	call	___chkstk_ms
	subq	%rax, %rsp
	.seh_stackalloc	65592
	.seh_endprologue
	cmpl	%r9d, %r8d
	movq	%rcx, %rbx
	movq	%rdx, %rdi
	movl	%r8d, %r12d
	movl	%r9d, %esi
	jb	.L118
	movl	(%rdx), %ebp
	leal	-1(%r9), %edx
	movl	$65536, %r8d
	leaq	48(%rsp), %rcx
	movzbl%dl, %edx
	call	memset
	movl	%esi, %r11d
	subl	$1, %r11d
	je	.L119
```

```
        xorl  %eax, %eax
        .p2align 4,,10
.L113:
        movzwl(%rdi,%rax), %edx
        movb  %al, 48(%rsp,%rdx)
        addq  $1, %rax
        cmpl  %eax, %r11d
        ja    .L113
.L112:
        leal  -4(%rsi), %r9d
        movl  %r12d, %r8d
        xorl  %edx, %edx
        leal  -3(%rsi), %eax
        shrl  $2, %r9d
        subl  %esi, %r8d
        leal  -2(%rsi), %r10d
        movslq%eax, %r14
        negq  %r9
        movl  %eax, 44(%rsp)
        leaq  -1(%r14), %r15
        salq  $2, %r9
        leaq  (%rdi,%r14), %r13
        jmp   .L117
        .p2align 4,,10
.L130:
        movl  %r10d, %eax
        subl  %ecx, %eax
        cmpl  %r10d, %ecx
        je    .L129
.L114:
        addl  %eax, %edx
        cmpl  %r8d, %edx
        ja    .L118
.L117:
        leal  (%rdx,%r10), %eax
        movzwl(%rbx,%rax), %eax
        movzbl48(%rsp,%rax), %ecx
        cmpl  %r11d, %ecx
        jne   .L130
        movl  %r11d, %eax
        addl  %eax, %edx
        cmpl  %r8d, %edx
        jbe   .L117
.L118:
        xorl  %eax, %eax
        jmp   .L128
        .p2align 4,,10
.L129:
        movl  %edx, %ecx
        movl  $1, %eax
        leaq  (%rbx,%rcx), %r12
        cmpl  (%r12), %ebp
        jne   .L114
        movl  44(%rsp), %esi
        testl %esi, %esi
        jle   .L124
        movl  (%r12,%r15), %esi
        cmpl  %esi, (%rdi,%r15)
        jne   .L114
        addq  %r14, %rcx
        xorl  %eax, %eax
        addq  %rbx, %rcx
        jmp   .L116
        .p2align 4,,10
.L132:
        movl  -5(%r13,%rax), %esi
        subq  $4, %rax
        cmpl  -1(%rcx,%rax), %esi
        jne   .L131
.L116:
        cmpq  %rax, %r9
        jne   .L132
.L124:
        movq  %r12, %rax
.L128:
        addq  $65592, %rsp
        popq  %rbx
        popq  %rsi
        popq  %rdi
        popq  %rbp
        popq  %r12
        popq  %r13
        popq  %r14
        popq  %r15
        ret
        .p2align 4,,10
.L131:
        movl  $1, %eax
        jmp   .L114
```

```
.L119:
       xorl  %r11d, %r11d
       jmp   .L112
*/

} //if (cbTarget<777)

             } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
       FoundAtPosition = cbPattern - ((4)-1) + 1;
       PRIMALpositionCANDIDATE=i;
       while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
             j = PRIMALpositionCANDIDATE + 1;
             while ( j <= (FoundAtPosition-1) ) {
                          if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                          j++;
             }
             PRIMALpositionCANDIDATE++;
       }
       PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
       if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
       if (cbPattern-i+1 <= PRIMALlength) break;
       if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                    PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                  PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                      PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                    PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                         PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                        PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                      PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                      PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                     PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;
```

```
                        00000000011111111112222222222333
                        12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
```

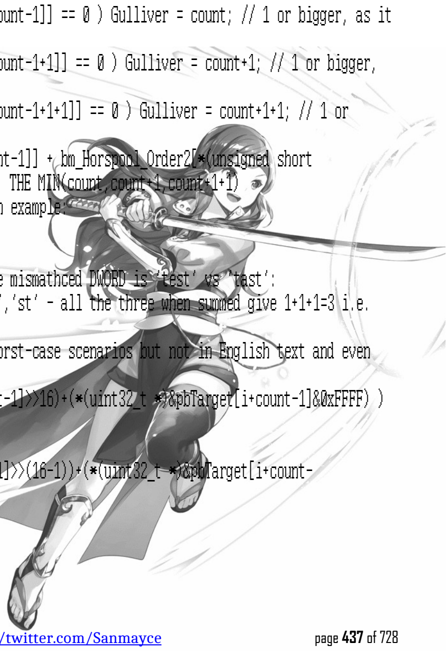


```
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
```

```
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00             mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
```

```
                        //"fast"
                        //"aste"
                        //"stes"
                        //"test"
                        //"est "
                        //"st f"
                        //"t fo"
                        //" fox"
                        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                        // Above line is replaced by next one with better hashing:
                        for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1;
                                    //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                                    // Above line is replaced by next one with better hashing:
                                    if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
      #1
                                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
      0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
      the middle DWORD is 'bc'.
                                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
      + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
      *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                // Above line is replaced by next one with better hashing:
                                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
      for ASCII texts with no symbols in range 128-255.
                                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                                // T           = 01010100
                                                // TT          = 01010100 01010100
                                                // TTTT        = 01010100 01010100 01010100 01010100
                                                // TTTT>>16    = 00000000 00000000 01010100 01010100
                                                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
      0 for English texts & 'ACGT' data.
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
      1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
      *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                // 'Maximus' uses branched 'if', again.
                                                if ( \
                                                ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
      1 )] ) == 0 \
                                                || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
      1 )] ) == 0 \
                                                ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                                // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//    DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//        DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//            DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                       |
//    DWORD #3                                                                       |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!  |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//        DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//            DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
      1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
      *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//        DWORD #2
//                DWORD #1
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
      + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                        // Order 4 [
                                                            // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                            // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
```

```
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                // Below comparison is UNIdirectional:
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }


                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,

    as it should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or

    bigger, as it should

                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.

    Gulliver remains 1.

                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even

    not in ACGT data, that's why I commented it in original 'Shockeroo'.

                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )

    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should

                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
```



```
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_6 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;

    uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
    uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
    uint32_t j, FoundAtPosition;

// Quadruplet [
  //char * pbTargetMax = pbTarget + cbTarget;
  //register unsigned long  ulHashPattern;
  uint32_t ulHashTarget;
  //unsigned long count;
  uint32_t countSTATIC;
  unsigned char SINGLET;
  uint32_t Quadruplet2nd;
  uint32_t Quadruplet3rd;
  uint32_t Quadruplet4th;
  uint32_t AdvanceHopperGrass;
// Quadruplet ]

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {
        // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
        // ...
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }
    } else { //if ( cbPattern<4 )
        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
     pbTarget = pbTarget+cbPattern;
     ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

  //SINGLET = *(char *)(pbPattern);
  SINGLET = ulHashPattern & 0xFF;
  Quadruplet2nd = SINGLET<<8;
  Quadruplet3rd = SINGLET<<16;
  Quadruplet4th = SINGLET<<24;

  for ( ;; )
  {
    AdvanceHopperGrass = 0;
    ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);
```

```
      if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//       count = countSTATIC;
//       while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//         if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//            count--;
//       }
       count = cbPattern-1;
       while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
           if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
            count--;
       }
       if ( count == 0) return((pbTarget-cbPattern));
      } else { // The goal here: to avoid memory accesses by stressing the registers.
   if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
       AdvanceHopperGrass++;
       if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
           AdvanceHopperGrass++;
           if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
       }
   }
     }

     AdvanceHopperGrass++;

     pbTarget = pbTarget + AdvanceHopperGrass;
      if (pbTarget > pbTargetMax)
         return(NULL);
   }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//   } else { //if ( cbPattern<4 )
//         if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                      // BMH order 2, needle should be >=4:
                      ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                      //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                      for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                      //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                      for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
     (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                      i=0;
                      while (i <= cbTarget-cbPattern) {
                                 Gulliver = 1; // 'Gulliver' is the skip
                                 //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                 if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                           //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                           if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
     == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
     below:
                                                                 count = cbPattern-4+1;
                                                                 while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                           count = count-4;
                                                                 if ( count <= 0 ) return(pbTarget+i);
                                                      }
                                           }
                                 } else Gulliver = cbPattern-(2-1);
                                 i = i + Gulliver;
                                 //GlobalI++; // Comment it, it is only for stats.
                      }
                      return(NULL);
//         } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                      // BMH order 2, needle should be >=4:
                      ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                    for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                      for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                      i=0;
                      while (i <= cbTarget-cbPattern) {
                                 Gulliver = 1; // 'Gulliver' is the skip
                                 if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                           if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
     below:
                                                                 count = cbPattern-4+1;
                                                                 while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                           count = count-4;
                                                                 if ( count <= 0 ) return(pbTarget+i);
                                                      }
                                           }
                                 } else Gulliver = cbPattern-(2-1);
                                 i = i + Gulliver;
                                 //GlobalI++; // Comment it, it is only for stats.
                      }
                      return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
```

```
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
      cbPattern = PRIMALlengthCANDIDATE;
      pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
      below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
      // The line below is BUGGY:
      //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
      // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
      //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
      if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
          if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
      below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
          }
```

```
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
---------------------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                         ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                                  ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don        ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                   ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe      ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence         ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                    ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf                ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40          add eax, 64
//  00142 66 0f 7f 44 14
//        60                movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00    cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70                movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00       movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00       movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2             mov edx, eax
//  00167 72 d6             jb .B5.21
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00                mov r8d, 65536
//  0012d 44 8b 26          mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff    call _intel_fast_memset
                // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
    it is okay to use 'memset'. !
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
                                        if ( count <= 0 ) return(pbTarget+i);
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
*/
```

```
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
```

```
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
        pushq %r15
        .seh_pushreg    %r15
        movl  $65592, %eax
        pushq %r14
        .seh_pushreg    %r14
        pushq %r13
        .seh_pushreg    %r13
        pushq %r12
        .seh_pushreg    %r12
        pushq %rbp
        .seh_pushreg    %rbp
        pushq %rdi
        .seh_pushreg    %rdi
        pushq %rsi
        .seh_pushreg    %rsi
        pushq %rbx
        .seh_pushreg    %rbx
        call  ___chkstk_ms
        subq  %rax, %rsp
        .seh_stackalloc    65592
```

```
	.seh_endprologue
	cmpl	%r9d, %r8d
	movq	%rcx, %rbx
	movq	%rdx, %rdi
	movl	%r8d, %r12d
	movl	%r9d, %esi
	jb	.L118
	movl	(%rdx), %ebp
	leal	-1(%r9), %edx
	movl	$65536, %r8d
	leaq	48(%rsp), %rcx
	movzbl%dl, %edx
	call	memset
	movl	%esi, %r11d
	subl	$1, %r11d
	je	.L119
	xorl	%eax, %eax
	.p2align 4,,10
.L113:
	movzwl(%rdi,%rax), %edx
	movb	%al, 48(%rsp,%rdx)
	addq	$1, %rax
	cmpl	%eax, %r11d
	ja	.L113
.L112:
	leal	-4(%rsi), %r9d
	movl	%r12d, %r8d
	xorl	%edx, %edx
	leal	-3(%rsi), %eax
	shrl	$2, %r9d
	subl	%esi, %r8d
	leal	-2(%rsi), %r10d
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl(%rbx,%rax), %eax
	movzbl48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
	cmpl	%r8d, %edx
	jbe	.L117
.L118:
	xorl	%eax, %eax
	jmp	.L128
	.p2align 4,,10
.L129:
	movl	%edx, %ecx
	movl	$1, %eax
	leaq	(%rbx,%rcx), %r12
	cmpl	(%r12), %ebp
	jne	.L114
	movl	44(%rsp), %esi
	testl	%esi, %esi
	jle	.L124
	movl	(%r12,%r15), %esi
	cmpl	%esi, (%rdi,%r15)
	jne	.L114
	addq	%r14, %rcx
	xorl	%eax, %eax
	addq	%rbx, %rcx
	jmp	.L116
	.p2align 4,,10
.L132:
	movl	-5(%r13,%rax), %esi
	subq	$4, %rax
	cmpl	-1(%rcx,%rax), %esi
	jne	.L131
.L116:
	cmpq	%rax, %r9
	jne	.L132
.L124:
```

```
        movq    %r12, %rax
.L128:
        addq    $65592, %rsp
        popq    %rbx
        popq    %rsi
        popq    %rdi
        popq    %rbp
        popq    %r12
        popq    %r13
        popq    %r14
        popq    %r15
        ret
        .p2align 4,,10
.L131:
        movl    $1, %eax
        jmp     .L114
.L119:
        xorl    %r11d, %r11d
        jmp     .L112
*/

} //if (cbTarget<777)

                } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                   PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                 PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                     PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                   PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                   PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                       PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                       PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                 PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                     PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
        while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
                j = PRIMALpositionCANDIDATE + 1;
                while ( j <= (FoundAtPosition-1) ) {
                                if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                                j++;
                }
                PRIMALpositionCANDIDATE++;
        }
        PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
        if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
        if (cbPattern-i+1 <= PRIMALlength) break;
        if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                   PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                 PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                     PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                   PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                        PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                       PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                     PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                    PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
```

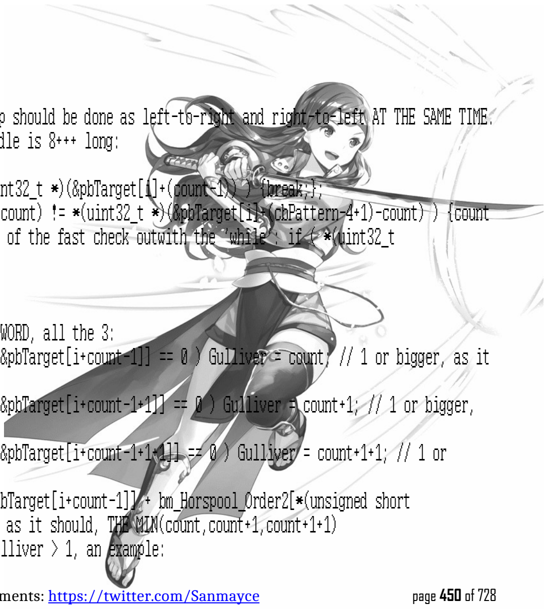
```
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       00000000011111111112222222222333
                       12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
```

```
Step 09_10: vvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
  if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
    is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvBOOMSHAKALAKAvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
  if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
  Result:
  PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
            if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
            if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                    count = cbPattern-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
} else {
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
```

```
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00            mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
```

```
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0))>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0))>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
    #1
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
    the middle DWORD is 'bc'.
                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // Above line is replaced by next one with better hashing:
                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
    for ASCII texts with no symbols in range 128-255.
                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                // T           = 01010100
                                // TT          = 01010100 01010100
                                // TTTT        = 01010100 01010100 01010100 01010100
                                // TTTT>>16    = 00000000 00000000 01010100 01010100
                                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
    0 for English texts & 'ACGT' data.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // 'Maximus' uses branched 'if', again.
                                if ( \
                                ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
    1 )] ) == 0 \
                                || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
    1 )] ) == 0 \
                                ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                         |
//     DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!    |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = -------------------------  |
//         DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!       |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ----------------------------  |
//             DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                  |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = -------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                         |
//     DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!    |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = -------------------------  |
//         DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!       |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ----------------------------  |
//             DWORD #1                                                                      |
```

```
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                  |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//          DWORD #2
//                  DWORD #1
                                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                        // Order 4 [
                                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                        count = cbPattern-4+1;
                                                        // Below comparison is UNIdirectional:
                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                        if ( count <= 0 ) return(pbTarget+i);
    }


                                                        // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                        // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                                      for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                              if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                              if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                                      }
//                                                      if ( count <= 0 ) return(pbTarget+i);
                                                                // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should

                                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
    as it should

                                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
    bigger, as it should
                                                        //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                                // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                                // Needle:    fastest tax
```



```
                                                                                    // Window: ...fastast tax...
                                                                                    // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                                                    // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
        Gulliver remains 1.
                                                                                    // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
        not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                                                    //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
        & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                                                    // Above line is replaced by next one with better hashing:
//                                                                                  if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
        1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                                        // Order 4 ]
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);

            } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
            } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_7 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        uint32_t ulHashPattern;
        signed long count;

        unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
        unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
        uint32_t i, Gulliver;

        uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
        uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
        uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

        if (cbPattern > cbTarget) return(NULL);

        if ( cbPattern<4 ) {
            // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
            // ...
            pbTarget = pbTarget+cbPattern;
            ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
            if ( cbPattern==3 ) {
                        for ( ;; ) {
                                    if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                    }
                                    if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                pbTarget++;
                                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                    }
                                    pbTarget++;
                                    if (pbTarget > pbTargetMax) return(NULL);
                        }
            } else {
            }
            for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
            }
        } else { //if ( cbPattern<4 )
            if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
```

```
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0¦1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0¦1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//     } else { //if ( cbPattern<4 )
//           if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                          // BMH order 2, needle should be >=4:
                          ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                          //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                          for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                          //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                          for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] ¦
       (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                          i=0;
                          while (i <= cbTarget-cbPattern) {
                                        Gulliver = 1; // 'Gulliver' is the skip
                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                        if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                                     //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                     if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
       == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                                  if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                                               count = cbPattern-4+1;
                                                                               while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                            count = count-4;
                                                                               if ( count <= 0 ) return(pbTarget+i);
                                                                  }
                                                     }
                                        } else Gulliver = cbPattern-(2-1);
                                        i = i + Gulliver;
                                        //GlobalI++; // Comment it, it is only for stats.
                          }
                          return(NULL);
//           } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                          // BMH order 2, needle should be >=4:
                          ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                          for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                          i=0;
                          while (i <= cbTarget-cbPattern) {
                                        Gulliver = 1; // 'Gulliver' is the skip
                                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                                     if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                                  if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
```

```
        below:
                                                                        count = cbPattern-4+1;
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                    count = count-4;
                                                                        if ( count <= 0 ) return(pbTarget+i);
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1);
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
        cbPattern = PRIMALlengthCANDIDATE;
        pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                                                                        count = cbPattern-4+1;
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                    count = count-4;
                                                                        if ( count <= 0 ) return(pbTarget+i);
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1);
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                        // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                        // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                        // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
        (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
```

```
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
-----------------------------------------------------------------------------------------------------
¦ testfile\Searcher                     ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦---------------------------------------------------------------------------------¦---------------------¦
¦ Compiler                              ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦---------------------------------------------------------------------------------¦---------------------¦
¦ The_Project_Gutenberg_EBook_of_Don    ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt               ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                       ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------¦---------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe  ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦        ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------¦---------------------¦
¦ Dragonfly_genome_shotgun_sequence     ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                       ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------¦---------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf            ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------¦---------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40          add eax, 64
//  00142 66 0f 7f 44 14
//        60                movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00    cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70                movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00       movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00       movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2             mov edx, eax
//  00167 72 d6             jb .B5.21
        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
        // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00                mov r8d, 65536
//  0012d 44 8b 26          mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff    call _intel_fast_memset
        // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
        // we must switch to 0¦1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
  it is okay to use 'memset'. !
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
            if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
  (Order-1) chars i.e. one char for Order 2
            if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
```

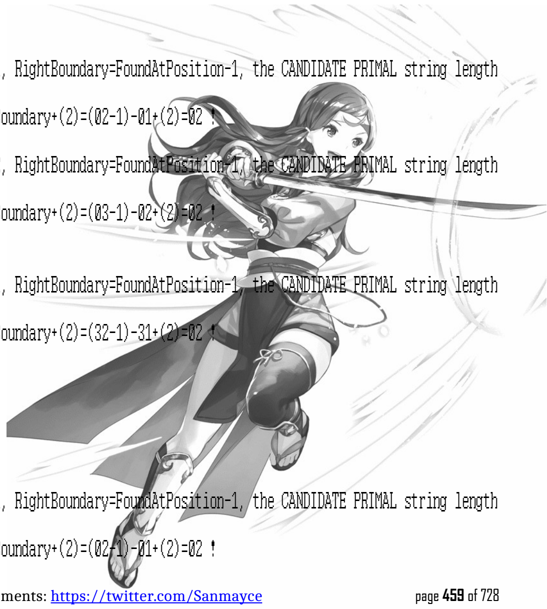
```
                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                    count = cbPattern-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                        count = count-4;
                    if ( count <= 0 ) return(pbTarget+i);
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
```

```
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
       pushq %r15
       .seh_pushreg      %r15
       movl  $65592, %eax
       pushq %r14
```

```
	.seh_pushreg	%r14
	pushq	%r13
	.seh_pushreg	%r13
	pushq	%r12
	.seh_pushreg	%r12
	pushq	%rbp
	.seh_pushreg	%rbp
	pushq	%rdi
	.seh_pushreg	%rdi
	pushq	%rsi
	.seh_pushreg	%rsi
	pushq	%rbx
	.seh_pushreg	%rbx
	call	___chkstk_ms
	subq	%rax, %rsp
	.seh_stackalloc	65592
	.seh_endprologue
	cmpl	%r9d, %r8d
	movq	%rcx, %rbx
	movq	%rdx, %rdi
	movl	%r8d, %r12d
	movl	%r9d, %esi
	jb	.L118
	movl	(%rdx), %ebp
	leal	-1(%r9), %edx
	movl	$65536, %r8d
	leaq	48(%rsp), %rcx
	movzbl%dl, %edx
	call	memset
	movl	%esi, %r11d
	subl	$1, %r11d
	je	.L119
	xorl	%eax, %eax
	.p2align 4,,10
.L113:
	movzwl(%rdi,%rax), %edx
	movb	%al, 48(%rsp,%rdx)
	addq	$1, %rax
	cmpl	%eax, %r11d
	ja	.L113
.L112:
	leal	-4(%rsi), %r9d
	movl	%r12d, %r8d
	xorl	%edx, %edx
	leal	-3(%rsi), %eax
	shrl	$2, %r9d
	subl	%esi, %r8d
	leal	-2(%rsi), %r10d
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl(%rbx,%rax), %eax
	movzbl48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
	cmpl	%r8d, %edx
	jbe	.L117
.L118:
	xorl	%eax, %eax
	jmp	.L128
	.p2align 4,,10
.L129:
	movl	%edx, %ecx
	movl	$1, %eax
	leaq	(%rbx,%rcx), %r12
	cmpl	(%r12), %ebp
	jne	.L114
	movl	44(%rsp), %esi
	testl %esi, %esi
	jle	.L124
	movl	(%r12,%r15), %esi
```

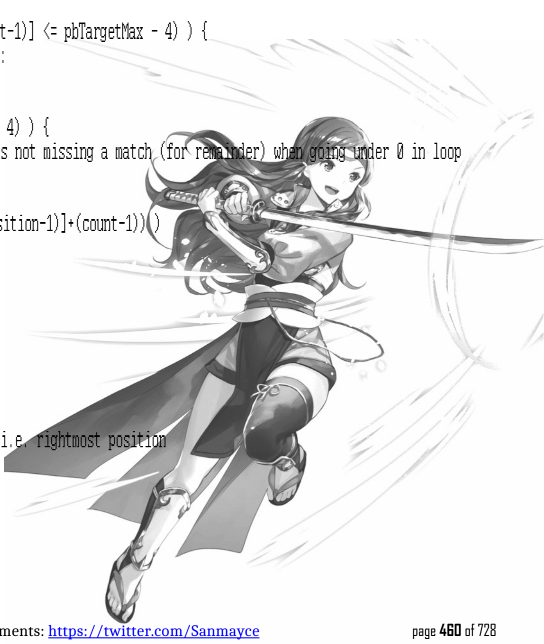
```
	cmpl	%esi, (%rdi,%r15)
	jne	.L114
	addq	%r14, %rcx
	xorl	%eax, %eax
	addq	%rbx, %rcx
	jmp	.L116
	.p2align 4,,10
.L132:
	movl	-5(%r13,%rax), %esi
	subq	$4, %rax
	cmpl	-1(%rcx,%rax), %esi
	jne	.L131
.L116:
	cmpq	%rax, %r9
	jne	.L132
.L124:
	movq	%r12, %rax
.L128:
	addq	$65592, %rsp
	popq	%rbx
	popq	%rsi
	popq	%rdi
	popq	%rbp
	popq	%r12
	popq	%r13
	popq	%r14
	popq	%r15
	ret
	.p2align 4,,10
.L131:
	movl	$1, %eax
	jmp	.L114
.L119:
	xorl	%r11d, %r11d
	jmp	.L112
*/

} //if (cbTarget<777)

		} else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
	FoundAtPosition = cbPattern - ((4)-1) + 1;
	PRIMALpositionCANDIDATE=i;
	while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
		j = PRIMALpositionCANDIDATE + 1;
		while ( j <= (FoundAtPosition-1) ) {
			if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
			j++;
		}
		PRIMALpositionCANDIDATE++;
	}
	PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
	if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
	if (cbPattern-i+1 <= PRIMALlength) break;
	if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                    PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                  PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                      PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                    PRIMALposition=02 PRIMALlength=08  'hazamish'
```

```
// Needle: Simplicius Simplicissimus                  PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                       PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                      PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                   PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                    PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                   PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       00000000011111111112222222222333
                       12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
```

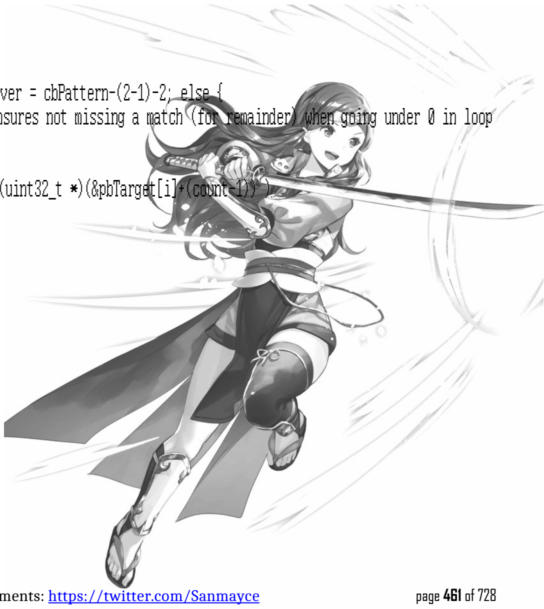


```
     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
```

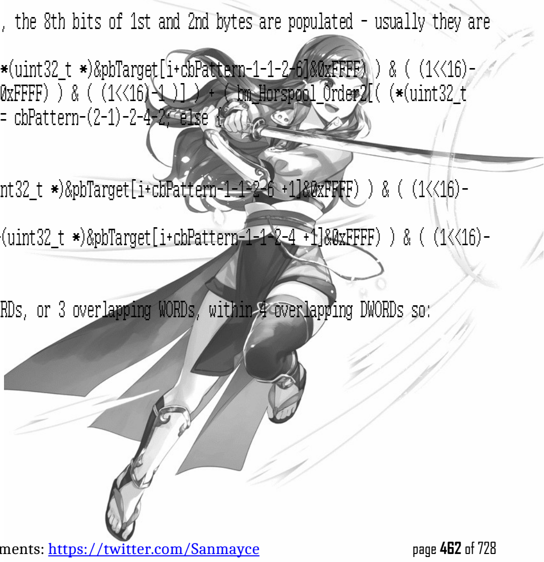
```
Step 02_00: {}v[vv]vvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
    is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
   if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
    is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
   if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
    is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
   if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
   Result:
   PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
```



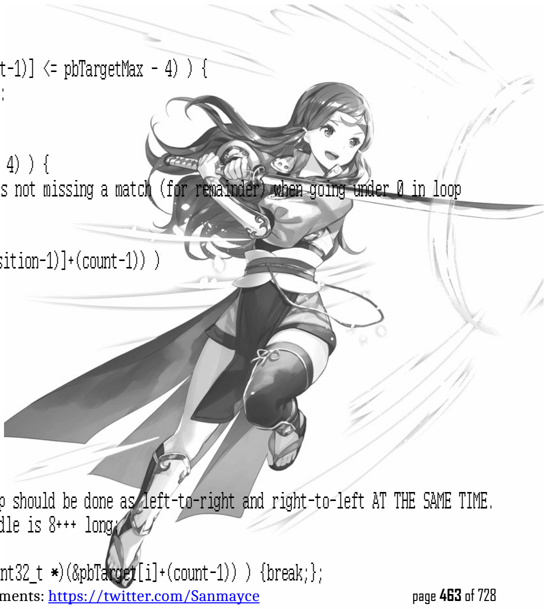
```
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00              mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
```



```
     //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
     // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
     //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
     if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
          if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
     below:
                    count = PRIMALlengthCANDIDATE-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                              count = count-4;
                    if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
          }
     }
}
     } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                            if ( count <= 0 ) return(pbTarget+i);
     }
                                                  }
                                        }
                              } else Gulliver = cbPattern-(2-1);
                              i = i + Gulliver;
                              //GlobalI++; // Comment it, it is only for stats.
                    }
                    return(NULL);

          } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                    // BMH pseudo-order 4, needle should be >=8+2:
                    ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                  for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                    // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
     example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //"fast"
                    //"aste"
                    //"stes"
                    //"test"
                    //"est "
                    //"st f"
                    //"t fo"
                    //" fox"
                    //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                    //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                    // Above line is replaced by next one with better hashing:
                    for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                    i=0;
                    while (i <= cbTarget-cbPattern) {
                              Gulliver = 1;
                              //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                              // Above line is replaced by next one with better hashing:
                              if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
     #1
                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
     0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                        // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
     the middle DWORD is 'bc'.
                                        // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
     + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
     *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                        // Above line is replaced by next one with better hashing:
                                        // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
     for ASCII texts with no symbols in range 128-255.
                                        // Example for genomesque pair TT+TT being shifted by (16-1):
                                        // T           = 01010100
                                        // TT          = 01010100 01010100
                                        // TTTT        = 01010100 01010100 01010100 01010100
                                        // TTTT>>16    = 00000000 00000000 01010100 01010100
                                        // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
     0 for English texts & 'ACGT' data.
                                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
     1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
     *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                        // 'Maximus' uses branched 'if', again.
                                        if ( \
                                        ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
     1 )] ) == 0 \

                                        || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
     1 )] ) == 0 \

                                        ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                        // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//    DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ----------------------- |
//       DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!    |<--
```



```
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = --------------------------- |
//            DWORD #1                                                                  |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                             |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                     |
//     DWORD #3                                                                    |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//        DWORD #2                                                                    |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = --------------------------- |
//            DWORD #1                                                                  |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                             |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
        1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
        *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//        DWORD #2
//                DWORD #1
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
        + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                        // Order 4 [
                                                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
        'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                                        count = cbPattern-4+1;
                                                                        // Below comparison is UNIdirectional:
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                count = count-4;

        if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
        // The line below is BUGGY:
        //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
        // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
        //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
        if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
                if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                        count = PRIMALlengthCANDIDATE-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                count = count-4;
                        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
                }
        }
}
        } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                                        if ( count <= 0 ) return(pbTarget+i);
        }


                                                                        // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                                        // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                                                      for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                                              if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
```



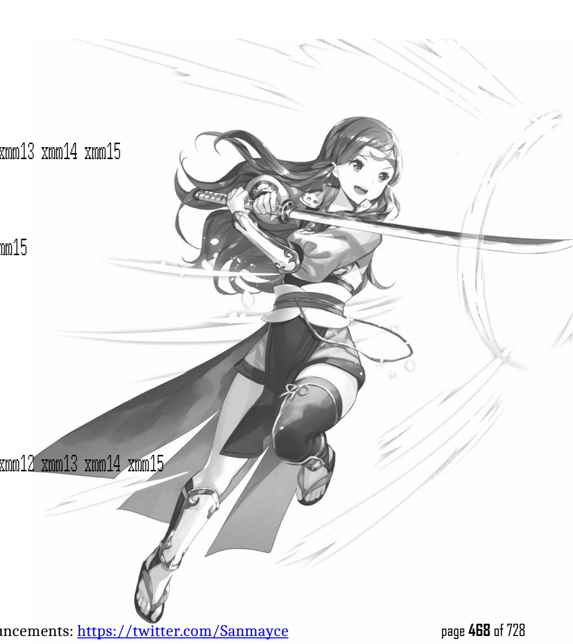
```
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
    as it should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
    Gulliver remains 1.

                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
    not in ACGT data, that's why I commented it in original 'Shockeroo'.

                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should

                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_8 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;

    uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
    uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
    uint32_t j, FoundAtPosition;

// Quadruplet [
  //char * pbTargetMax = pbTarget + cbTarget;
  //register unsigned long  ulHashPattern;
  uint32_t ulHashTarget;
  //unsigned long count;
  uint32_t countSTATIC;
  unsigned char SINGLET;
  uint32_t Quadruplet2nd;
  uint32_t Quadruplet3rd;
  uint32_t Quadruplet4th;
  uint32_t AdvanceHopperGrass;
// Quadruplet ]

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {
        // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
        // ...
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
```

```
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }
    } else { //if ( cbPattern<4 )
        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
     pbTarget = pbTarget+cbPattern;
     ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

  //SINGLET = *(char *)(pbPattern);
  SINGLET = ulHashPattern & 0xFF;
  Quadruplet2nd = SINGLET<<8;
  Quadruplet3rd = SINGLET<<16;
  Quadruplet4th = SINGLET<<24;

  for ( ;; )
  {
    AdvanceHopperGrass = 0;
    ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

     if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//           if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
      count = cbPattern-1;
      while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
         if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
          count--;
      }
      if ( count == 0) return((pbTarget-cbPattern));
     } else { // The goal here: to avoid memory accesses by stressing the registers.
  if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
      AdvanceHopperGrass++;
      if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
      }
  }
    }

    AdvanceHopperGrass++;

    pbTarget = pbTarget + AdvanceHopperGrass;
     if (pbTarget > pbTargetMax)
       return(NULL);
  }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//    } else { //if ( cbPattern<4 )
//        if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
    (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                        if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
    == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
```

```
				i = i + Gulliver;
				//GlobalI++; // Comment it, it is only for stats.
			}
			return(NULL);
//		} else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
			// BMH order 2, needle should be >=4:
			ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//			for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
			for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
			i=0;
			while (i <= cbTarget-cbPattern) {
				Gulliver = 1; // 'Gulliver' is the skip
				if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
					if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
						if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
							count = cbPattern-4+1; 
							while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
								count = count-4;
							if ( count <= 0 ) return(pbTarget+i);
						}
					}
				} else Gulliver = cbPattern-(2-1);
				i = i + Gulliver;
				//GlobalI++; // Comment it, it is only for stats.
			}
			return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
       cbPattern = PRIMALlengthCANDIDATE;
       pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
			// BMH order 2, needle should be >=4:
			ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
			for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
			for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
			i=0;
			while (i <= cbTarget-cbPattern) {
				Gulliver = 1; // 'Gulliver' is the skip
				if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
					if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
						if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
							count = cbPattern-4+1; 
							while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
								count = count-4;
							if ( count <= 0 ) return(pbTarget+i);
						}
					}
				} else Gulliver = cbPattern-(2-1);
				i = i + Gulliver;
				//GlobalI++; // Comment it, it is only for stats.
			}
			return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
			ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
			for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
			// The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
			// The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
			// The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
			// The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
			//memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
			for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
			i=0;
			while (i <= cbTarget-cbPattern) {
				Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
```

```
                                       if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
       (Order-1) chars i.e. one char for Order 2
                                       if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                 count = cbPattern-4+1;
                                                                 while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                              count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
       // The line below is BUGGY:
       //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
       // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
       //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
       if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
             if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                          count = PRIMALlengthCANDIDATE-4+1;
                          while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                       count = count-4;
                          if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
             }
       }
}
                                                    }
                                                    Gulliver = 1;
                                       } else
                                                    Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                       }
                                       i = i + Gulliver;
                                       //GlobalI++; // Comment it, it is only for stats.
                          }
                          return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
---------------------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                         ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
|-------------------------------------------------------------------------------------------|-----------------------|
¦ Compiler                                  ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
|-------------------------------------------------------------------------------------------|-----------------------|
¦ The_Project_Gutenberg_EBook_of_Don        ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                   ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
|-------------------------------------------------------------------------------------------|-----------------------|
¦ The_Project_Gutenberg_EBook_of_Dokoe      ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
|-------------------------------------------------------------------------------------------|-----------------------|
¦ Dragonfly_genome_shotgun_sequence         ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                    ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
|-------------------------------------------------------------------------------------------|-----------------------|
¦ LAOTZU_Wu_Wei_(BINARY).pdf                ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
|-------------------------------------------------------------------------------------------|-----------------------|
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                          ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                          for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                          // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
```

```
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00            mov r8d, 65536
//  0012d 44 8b 26      mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                        // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                        // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
      it is okay to use 'memset'. !
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
                                                            if ( count <= 0 ) return(pbTarget+i);
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
```



```
        cmp     r10d, r15d                                    ;3515.4
        movzx   ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov     BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb      .B6.8       ; Prob 64%                        ;3515.4
                            ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                     ; Preds .B6.8
        lea     eax, DWORD PTR [1+r10+r10]                    ;3515.4
                            ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                    ; Preds .B6.9 .B6.6
        dec     eax                                           ;3515.36
        cmp     eax, r11d                                     ;3515.4
        jae     .B6.12      ; Prob 15%                        ;3515.4
                            ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                    ; Preds .B6.10
        movzx   r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov     BYTE PTR [48+rsp+r10], al                     ;3515.36
                            ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                    ; Preds .B6.5 .B6.10 .B6.11
        xor     r10d, r10d                                    ;3516.4
        lea     r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd  r15, r15d                                     ;3522.7
        sub     r8d, r9d                                      ;3517.16
        lea     r11d, DWORD PTR [-2+r9]                       ;3520.32
                            ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                    ; Preds .B6.12 .B6.24
        lea     eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx   ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx   eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp     eax, ebp                                      ;3519.32
        je      .B6.24      ; Prob 50%                        ;3519.32
                            ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                    ; Preds .B6.13
        cmp     eax, r11d                                     ;3520.32
        jne     .B6.23      ; Prob 62%                        ;3520.32
                            ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                    ; Preds .B6.14
        mov     eax, r10d                                     ;3521.25
        add     rax, rcx                                      ;3521.25
        cmp     r13d, DWORD PTR [rax]                         ;3521.40
        je      .B6.17      ; Prob 50%                        ;3521.40
                            ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                    ; Preds .B6.26 .B6.15
        mov     eax, 1                                        ;3527.6
        jmp     .B6.24      ; Prob 100%                       ;3527.6
                            ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                    ; Preds .B6.15
        mov     rbx, r15                                      ;3522.7
        test    r15, r15                                      ;3523.23
        jle     .B6.22      ; Prob 2%                         ;3523.23
                            ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                    ; Preds .B6.17
        mov     QWORD PTR [32+rsp], rsi                       ;
                            ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                    ; Preds .B6.20 .B6.18
        mov     esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp     esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne     .B6.26      ; Prob 20%                        ;3523.79
                            ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                    ; Preds .B6.19
        add     rbx, -4                                       ;3524.22
        test    rbx, rbx                                      ;3523.23
        jg      .B6.19      ; Prob 82%                        ;3523.23
                            ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                    ; Preds .B6.20
        mov     rsi, QWORD PTR [32+rsp]                       ;
                            ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                    ; Preds .B6.17 .B6.21
        add     rsp, 65592                                    ;3525.32
        pop     rbp                                           ;3525.32
        pop     r15                                           ;3525.32
        pop     r13                                           ;3525.32
        pop     rbx                                           ;3525.32
        ret                                                   ;3525.32
                            ; LOE
.B6.23::                    ; Preds .B6.14
        neg     eax                                           ;3529.17
        add     eax, r9d                                      ;3529.17
        add     eax, -2                                       ;3529.40
                            ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                    ; Preds .B6.16 .B6.23 .B6.13
        add     r10d, eax                                     ;3531.13
        cmp     r10d, r8d                                     ;3517.25
        jbe     .B6.13      ; Prob 82%                        ;3517.25
                            ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                    ; Preds .B6.1 .B6.24
        xor     eax, eax                                      ;3534.10
        add     rsp, 65592                                    ;3534.10
        pop     rbp                                           ;3534.10
        pop     r15                                           ;3534.10
```

```
        pop       r13                                     ;3534.10
        pop       rbx                                     ;3534.10
        ret                                               ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19            ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                 ;
        jmp       .B6.16        ; Prob 100%               ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
	pushq	%r15
	.seh_pushreg	%r15
	movl	$65592, %eax
	pushq	%r14
	.seh_pushreg	%r14
	pushq	%r13
	.seh_pushreg	%r13
	pushq	%r12
	.seh_pushreg	%r12
	pushq	%rbp
	.seh_pushreg	%rbp
	pushq	%rdi
	.seh_pushreg	%rdi
	pushq	%rsi
	.seh_pushreg	%rsi
	pushq	%rbx
	.seh_pushreg	%rbx
	call	___chkstk_ms
	subq	%rax, %rsp
	.seh_stackalloc	65592
	.seh_endprologue
	cmpl	%r9d, %r8d
	movq	%rcx, %rbx
	movq	%rdx, %rdi
	movl	%r8d, %r12d
	movl	%r9d, %esi
	jb	.L118
	movl	(%rdx), %ebp
	leal	-1(%r9), %edx
	movl	$65536, %r8d
	leaq	48(%rsp), %rcx
	movzbl%dl, %edx
	call	memset
	movl	%esi, %r11d
	subl	$1, %r11d
	je	.L119
	xorl	%eax, %eax
	.p2align 4,,10
.L113:
	movzwl(%rdi,%rax), %edx
	movb	%al, 48(%rsp,%rdx)
	addq	$1, %rax
	cmpl	%eax, %r11d
	ja	.L113
.L112:
	leal	-4(%rsi), %r9d
	movl	%r12d, %r8d
	xorl	%edx, %edx
	leal	-3(%rsi), %eax
	shrl	$2, %r9d
	subl	%esi, %r8d
	leal	-2(%rsi), %r10d
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl(%rbx,%rax), %eax
	movzbl48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
```

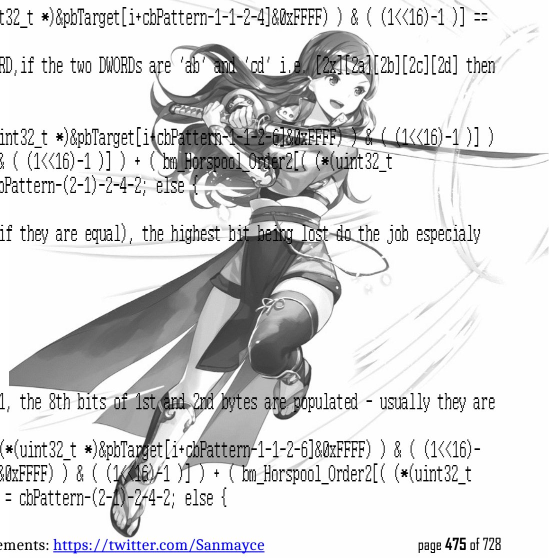
```
        cmpl  %r8d, %edx
        jbe   .L117
.L118:
        xorl  %eax, %eax
        jmp   .L128
        .p2align 4,,10
.L129:
        movl  %edx, %ecx
        movl  $1, %eax
        leaq  (%rbx,%rcx), %r12
        cmpl  (%r12), %ebp
        jne   .L114
        movl  44(%rsp), %esi
        testl %esi, %esi
        jle   .L124
        movl  (%r12,%r15), %esi
        cmpl  %esi, (%rdi,%r15)
        jne   .L114
        addq  %r14, %rcx
        xorl  %eax, %eax
        addq  %rbx, %rcx
        jmp   .L116
        .p2align 4,,10
.L132:
        movl  -5(%r13,%rax), %esi
        subq  $4, %rax
        cmpl  -1(%rcx,%rax), %esi
        jne   .L131
.L116:
        cmpq  %rax, %r9
        jne   .L132
.L124:
        movq  %r12, %rax
.L128:
        addq  $65592, %rsp
        popq  %rbx
        popq  %rsi
        popq  %rdi
        popq  %rbp
        popq  %r12
        popq  %r13
        popq  %r14
        popq  %r15
        ret
        .p2align 4,,10
.L131:
        movl  $1, %eax
        jmp   .L114
.L119:
        xorl  %r11d, %r11d
        jmp   .L112
*/

} //if (cbTarget<777)

                } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv          PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
        while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
                j = PRIMALpositionCANDIDATE + 1;
                while ( j <= (FoundAtPosition-1) ) {
                                if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                                j++;
                }
                PRIMALpositionCANDIDATE++;
        }
        PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
```

```
        if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
        if (cbPattern-i+1 <= PRIMALlength) break;
        if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                   PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                 PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                     PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                   PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                        PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                       PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                     PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                    PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       000000000111111111122222222223333
                       123456789012345678901234567890123
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
```

```
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
       (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
       // The line below is BUGGY:
       //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
       // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
       //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
       if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
            if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                        count = PRIMALlengthCANDIDATE-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
```

```
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//      00      mov r8d, 65536
//  00452 44 89 5c 24 20    mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60    mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff    call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
```

```
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
    #1
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
    the middle DWORD is 'bc'.
                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // Above line is replaced by next one with better hashing:
                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
    for ASCII texts with no symbols in range 128-255.
                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                // T           = 01010100
                                // TT          = 01010100 01010100
                                // TTTT        = 01010100 01010100 01010100 01010100
                                // TTTT>>16    = 00000000 00000000 01010100 01010100
                                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
    0 for English texts & 'ACGT' data.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // 'Maximus' uses branched 'if', again.
```



```
                                                      if ( \
                                                      ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
        1 )] ) == 0 \
                                                      || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
        1 )] ) == 0 \
                                                      ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                                      // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//    DWORD #3                                                                         |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//        DWORD #2                                                                        |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//            DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//    DWORD #3                                                                         |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//        DWORD #2                                                                        |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//            DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
        1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
        *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//        DWORD #2
//                DWORD #1
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
        + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                                   if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                               // Order 4 [
                                                                   // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                                   // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
        'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                                   //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                                   //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                                   //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                                   //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                                   //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                                   //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                                   //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                                   //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                                               count = cbPattern-4+1;
                                                                               // Below comparison is UNIdirectional:
                                                                               while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                           count = count-4;

        if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
        // The line below is BUGGY:
        //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
        // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
        //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
        if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
            if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
```

```
        below:
                                count = PRIMALlengthCANDIDATE-4+1;
                                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                                count = count-4;
                                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
                }
        }
}
        } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                                                                if ( count <= 0 ) return(pbTarget+i);
        }


                                                                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                                                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                                                                              if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                                                                              if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
        = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
        *)&pbTarget[i] == ulHashPattern)
//                                                                                              }
//                                                                                              if ( count <= 0 ) return(pbTarget+i);
                                                                                                                // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                                                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
        should

                                                                                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
        as it should

                                                                                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
        bigger, as it should

                                                                                                //              if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
        *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                                                                                // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                                                                                // Needle:    fastest tax
                                                                                                                // Window: ...fastast tax...
                                                                                                                // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                                                                                // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
        Gulliver remains 1.

                                                                                                                // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
        not in ACGT data, that's why I commented it in original 'Shockeroo'.

                                                                                                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
        & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should

                                                                                                                // Above line is replaced by next one with better hashing:
//                                                                                                              if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
        1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                                                                // Order 4 ]
                                                                                }
                                                                }
                                                } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                                                i = i + Gulliver;
                                                //GlobalI++; // Comment it, it is only for stats.
                                }
                                return(NULL);

                } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
                } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_9 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        uint32_t ulHashPattern;
        signed long count;

        unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
        unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
        uint32_t i, Gulliver;

        uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
        uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
        uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

        if (cbPattern > cbTarget) return(NULL);

        if ( cbPattern<4 ) {
                // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
                // ...
```

```
          pbTarget = pbTarget+cbPattern;
          ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
          if ( cbPattern==3 ) {
                    for ( ;; ) {
                              if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                        if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                              }
                              if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                        pbTarget++;
                                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                              }
                              pbTarget++;
                              if (pbTarget > pbTargetMax) return(NULL);
                    }
          } else {
          }
          for ( ;; ) {
                    if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                    if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                    pbTarget++;
                    if (pbTarget > pbTargetMax) return(NULL);
          }
     } else { //if ( cbPattern<4 )
          if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
      pbTarget = pbTarget+cbPattern;
      ulHashPattern = *(uint32_t *)(pbPattern);
//       countSTATIC = cbPattern-1;

   //SINGLET = *(char *)(pbPattern);
   SINGLET = ulHashPattern & 0xFF;
   Quadruplet2nd = SINGLET<<8;
   Quadruplet3rd = SINGLET<<16;
   Quadruplet4th = SINGLET<<24;

   for ( ;; )
   {
     AdvanceHopperGrass = 0;
     ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

      if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//        count = countSTATIC;
//        while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//           if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//             count--;
//        }
       count = cbPattern-1;
       while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
           if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
            count--;
       }
       if ( count == 0) return((pbTarget-cbPattern));
      } else { // The goal here: to avoid memory accesses by stressing the registers.
   if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
       AdvanceHopperGrass++;
       if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
           AdvanceHopperGrass++;
           if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
       }
   }
     }

     AdvanceHopperGrass++;

     pbTarget = pbTarget + AdvanceHopperGrass;
      if (pbTarget > pbTargetMax)
         return(NULL);
   }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//     } else { //if ( cbPattern<4 )
//          if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                    // BMH order 2, needle should be >=4:
                    ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                    //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                    for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                    //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                    for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
     (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                    i=0;
```

```
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                        if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
    == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
//              } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

// Slower than Swampshine's simple 0¦1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
    cbPattern = PRIMALlengthCANDIDATE;
    pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
```

```
                                        //GlobalI++; // Comment it, it is only for stats.
                                }
                                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                                i=0;
                                while (i <= cbTarget-cbPattern) {
                                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                        count = cbPattern-4+1;
                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                                }
                                                Gulliver = 1;
                                        } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                        }
                                        i = i + Gulliver;
                                        //GlobalI++; // Comment it, it is only for stats.
                                }
                                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
---------------------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                         ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                                  ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don        ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                   ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe      ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence         ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                    ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf                ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦-------------------------------------------------------------------------------------------¦-----------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
```

```
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00              mov r8d, 65536
//  0012d 44 8b 26        mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                        // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                        // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
      it is okay to use 'memset'. !
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
                                                            if ( count <= 0 ) return(pbTarget+i);
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
```

```
                               ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                        ; Preds .B6.5
        mov       eax, 1                                      ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                     ;3515.4
        mov       r15d, r11d                                  ;3515.4
        xor       r10d, r10d                                  ;3515.4
        shr       r15d, 1                                     ;3515.4
        test      r15d, r15d                                  ;3515.4
        jbe       .B6.10        ; Prob 15%                    ;3515.4
                               ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                        ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                    ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                     ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                   ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                  ;3515.36
        inc       r10d                                        ;3515.4
        cmp       r10d, r15d                                  ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                     ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                   ;3515.36
        jb        .B6.8         ; Prob 64%                    ;3515.4
                               ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                        ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                  ;3515.4
                               ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                       ; Preds .B6.9 .B6.6
        dec       eax                                         ;3515.36
        cmp       eax, r11d                                   ;3515.4
        jae       .B6.12        ; Prob 15%                    ;3515.4
                               ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                       ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                    ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                   ;3515.36
                               ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                       ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                  ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                     ;3522.27
        movsxd    r15, r15d                                   ;3522.7
        sub       r8d, r9d                                    ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                     ;3520.32
                               ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                       ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                  ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                     ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                  ;3518.16
        cmp       eax, ebp                                    ;3519.32
        je        .B6.24        ; Prob 50%                    ;3519.32
                               ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                       ; Preds .B6.13
        cmp       eax, r11d                                   ;3520.32
        jne       .B6.23        ; Prob 62%                    ;3520.32
                               ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                       ; Preds .B6.14
        mov       eax, r10d                                   ;3521.25
        add       rax, rcx                                    ;3521.25
        cmp       r13d, DWORD PTR [rax]                       ;3521.40
        je        .B6.17        ; Prob 50%                    ;3521.40
                               ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                       ; Preds .B6.26 .B6.15
        mov       eax, 1                                      ;3527.6
        jmp       .B6.24        ; Prob 100%                   ;3527.6
                               ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                       ; Preds .B6.15
        mov       rbx, r15                                    ;3522.7
        test      r15, r15                                    ;3523.23
        jle       .B6.22        ; Prob 2%                     ;3523.23
                               ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                       ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                     ;
                               ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                       ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                 ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                 ;3523.79
        jne       .B6.26        ; Prob 20%                    ;3523.79
                               ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                       ; Preds .B6.19
        add       rbx, -4                                     ;3524.22
        test      rbx, rbx                                    ;3523.23
        jg        .B6.19        ; Prob 82%                    ;3523.23
                               ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                       ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                     ;
                               ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                       ; Preds .B6.17 .B6.21
        add       rsp, 65592                                  ;3525.32
        pop       rbp                                         ;3525.32
        pop       r15                                         ;3525.32
        pop       r13                                         ;3525.32
        pop       rbx                                         ;3525.32
        ret                                                   ;3525.32
```

```
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
       pushq %r15
       .seh_pushreg      %r15
       movl  $65592, %eax
       pushq %r14
       .seh_pushreg      %r14
       pushq %r13
       .seh_pushreg      %r13
       pushq %r12
       .seh_pushreg      %r12
       pushq %rbp
       .seh_pushreg      %rbp
       pushq %rdi
       .seh_pushreg      %rdi
       pushq %rsi
       .seh_pushreg      %rsi
       pushq %rbx
       .seh_pushreg      %rbx
       call  ___chkstk_ms
       subq  %rax, %rsp
       .seh_stackalloc   65592
       .seh_endprologue
       cmpl  %r9d, %r8d
       movq  %rcx, %rbx
       movq  %rdx, %rdi
       movl  %r8d, %r12d
       movl  %r9d, %esi
       jb    .L118
       movl  (%rdx), %ebp
       leal  -1(%r9), %edx
       movl  $65536, %r8d
       leaq  48(%rsp), %rcx
       movzbl%dl, %edx
       call  memset
       movl  %esi, %r11d
       subl  $1, %r11d
       je    .L119
       xorl  %eax, %eax
       .p2align 4,,10
.L113:
       movzwl(%rdi,%rax), %edx
       movb  %al, 48(%rsp,%rdx)
       addq  $1, %rax
       cmpl  %eax, %r11d
       ja    .L113
.L112:
       leal  -4(%rsi), %r9d
       movl  %r12d, %r8d
       xorl  %edx, %edx
       leal  -3(%rsi), %eax
       shrl  $2, %r9d
       subl  %esi, %r8d
       leal  -2(%rsi), %r10d
       movslq%eax, %r14
       negq  %r9
       movl  %eax, 44(%rsp)
       leaq  -1(%r14), %r15
       salq  $2, %r9
       leaq  (%rdi,%r14), %r13
       jmp   .L117
       .p2align 4,,10
.L130:
```

```
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl	(%rbx,%rax), %eax
	movzbl	48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
	cmpl	%r8d, %edx
	jbe	.L117
.L118:
	xorl	%eax, %eax
	jmp	.L128
	.p2align 4,,10
.L129:
	movl	%edx, %ecx
	movl	$1, %eax
	leaq	(%rbx,%rcx), %r12
	cmpl	(%r12), %ebp
	jne	.L114
	movl	44(%rsp), %esi
	testl	%esi, %esi
	jle	.L124
	movl	(%r12,%r15), %esi
	cmpl	%esi, (%rdi,%r15)
	jne	.L114
	addq	%r14, %rcx
	xorl	%eax, %eax
	addq	%rbx, %rcx
	jmp	.L116
	.p2align 4,,10
.L132:
	movl	-5(%r13,%rax), %esi
	subq	$4, %rax
	cmpl	-1(%rcx,%rax), %esi
	jne	.L131
.L116:
	cmpq	%rax, %r9
	jne	.L132
.L124:
	movq	%r12, %rax
.L128:
	addq	$65592, %rsp
	popq	%rbx
	popq	%rsi
	popq	%rdi
	popq	%rbp
	popq	%r12
	popq	%r13
	popq	%r14
	popq	%r15
	ret
	.p2align 4,,10
.L131:
	movl	$1, %eax
	jmp	.L114
.L119:
	xorl	%r11d, %r11d
	jmp	.L112
*/

} //if (cbTarget<777)

		} else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                   PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                 PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                     PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                   PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                   PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                       PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                       PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                 PRIMALposition=03 PRIMALlength=09  'key-likey'
```

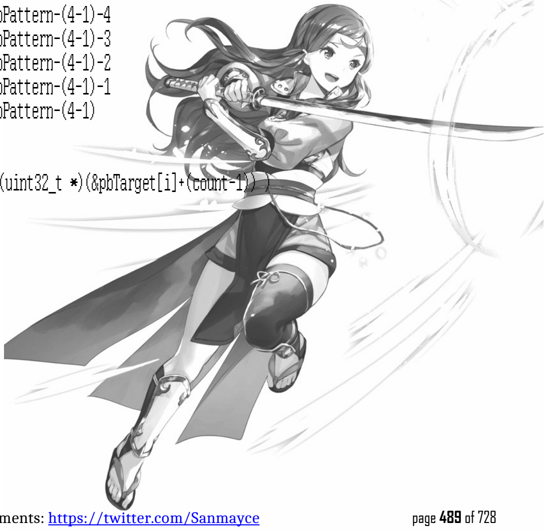
```
// Needle: BOOOOOM                                    PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                               PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                               PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
        while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
              j = PRIMALpositionCANDIDATE + 1;
              while ( j <= (FoundAtPosition-1) ) {
                            if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                            j++;
              }
              PRIMALpositionCANDIDATE++;
        }
        PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
        if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
        if (cbPattern-i+1 <= PRIMALlength) break;
        if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv          PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv          PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv          PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                  PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                            PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                    PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                   PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                  PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                  PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                       PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                      PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                   PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                    PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                   PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                        00000000011111111112222222222333
                        12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
```

```
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
    Result:
    PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'


    PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
    Result:
    PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'


    PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
    Result:
    PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
```

```
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
} else {
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
}
*/
// Revision 2 commented section ]

    if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
```

```
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00                mov r8d, 65536
//  00452 44 89 5c 24 20    mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60    mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff    call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
    #1
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
    the middle DWORD is 'bc'.
                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
```

```
        + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
        *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                // Above line is replaced by next one with better hashing:
                                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
        for ASCII texts with no symbols in range 128-255.
                                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                                // T           = 01010100
                                                // TT          = 01010100 01010100
                                                // TTTT        = 01010100 01010100 01010100 01010100
                                                // TTTT>>16    = 00000000 00000000 01010100 01010100
                                                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
        0 for English texts & 'ACGT' data.
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
        1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
        *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                // 'Maximus' uses branched 'if', again.
                                                if ( \
                                                ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
        1 )] ) == 0 \
                                                || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
        1 )] ) == 0 \
                                                ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                                // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                       |
//     DWORD #3                                                                     |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!  |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                     |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                    |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                       |
//     DWORD #3                                                                     |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!  |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                     |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                    |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
        1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
        *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
                                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
        + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                        // Order 4 [
                                                                // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                                // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
        'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                                //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                                //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                                //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                                //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                                //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                                //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                                //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                                //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                                        count = cbPattern-4+1;
                                                                        // Below comparison is UNIdirectional:
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                count = count-4;

        if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
```



```
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }


                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,

    as it should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or

    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.

    Gulliver remains 1.

                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even

    not in ACGT data, that's why I commented it in original 'Shockeroo'.

                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )

    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should

                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_0 (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;

    uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
    uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
    uint32_t j, FoundAtPosition;

// Quadruplet [
  //char * pbTargetMax = pbTarget + cbTarget;
```

```
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

       if (cbPattern > cbTarget) return(NULL);

       if ( cbPattern<4 ) {
             // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
             // ...
             pbTarget = pbTarget+cbPattern;
             ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
             if ( cbPattern==3 ) {
                           for ( ;; ) {
                                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                     if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                        }
                                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                     pbTarget++;
                                                     if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                        }
                                        pbTarget++;
                                        if (pbTarget > pbTargetMax) return(NULL);
                           }
             } else {
             }
             for ( ;; ) {
                           if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                           if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                           pbTarget++;
                           if (pbTarget > pbTargetMax) return(NULL);
             }
       } else { //if ( cbPattern<4 )
             if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
```

```
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//     } else { //if ( cbPattern<4 )
//      if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] | (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                        if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) ) == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
//      } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
        cbPattern = PRIMALlengthCANDIDATE;
        pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
```

```
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
--------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                      ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦---------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                               ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦---------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don     ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                        ¦            ¦          ¦            ¦          ¦            ¦          ¦
```

```
¦-------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe    ¦       582 ¦      760 ¦       3094 ¦      2898 ¦        2410 ¦      3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦           ¦          ¦            ¦           ¦             ¦           ¦
¦ 899,425 bytes                           ¦           ¦          ¦            ¦           ¦             ¦           ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence       ¦       104 ¦      109 ¦        445 ¦       458 ¦         484 ¦       553 ¦
¦ _(ACGT_alphabet).fasta                  ¦           ¦          ¦            ¦           ¦             ¦           ¦
¦ 4,487,433 bytes                         ¦           ¦          ¦            ¦           ¦             ¦           ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf              ¦        99 ¦      144 ¦        629 ¦       580 ¦         185 ¦       570 ¦
¦ 954,035 bytes                           ¦           ¦          ¦            ¦           ¦             ¦           ¦
¦-------------------------------------------------------------------------------------¦-----------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
			ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
			for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
			// The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
			//memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
			// The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00              mov r8d, 65536
//  0012d 44 8b 26        mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
			// ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
			// we must switch to 0¦1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
	it is okay to use 'memset'. !
			for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
			i=0;
			while (i <= cbTarget-cbPattern) {
				Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
				if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
	(Order-1) chars i.e. one char for Order 2
				if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
					if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
						count = cbPattern-4+1;
						while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
							count = count-4;
						if ( count <= 0 ) return(pbTarget+i);
					}
					Gulliver = 1;
				} else
					Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
				}
				i = i + Gulliver;
				//GlobalI++; // Comment it, it is only for stats.
			}
			return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
```

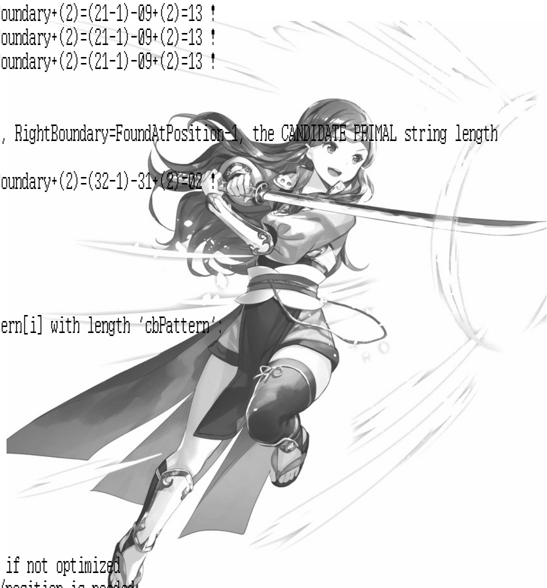
```
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
```

```
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
       pushq %r15
       .seh_pushreg       %r15
       movl  $65592, %eax
       pushq %r14
       .seh_pushreg       %r14
       pushq %r13
       .seh_pushreg       %r13
       pushq %r12
       .seh_pushreg       %r12
       pushq %rbp
       .seh_pushreg       %rbp
       pushq %rdi
       .seh_pushreg       %rdi
       pushq %rsi
       .seh_pushreg       %rsi
       pushq %rbx
       .seh_pushreg       %rbx
       call  ___chkstk_ms
       subq  %rax, %rsp
       .seh_stackalloc    65592
       .seh_endprologue
       cmpl  %r9d, %r8d
       movq  %rcx, %rbx
       movq  %rdx, %rdi
       movl  %r8d, %r12d
       movl  %r9d, %esi
       jb    .L118
       movl  (%rdx), %ebp
       leal  -1(%r9), %edx
       movl  $65536, %r8d
       leaq  48(%rsp), %rcx
       movzbl%dl, %edx
       call  memset
       movl  %esi, %r11d
       subl  $1, %r11d
       je    .L119
       xorl  %eax, %eax
       .p2align 4,,10
.L113:
       movzwl(%rdi,%rax), %edx
       movb  %al, 48(%rsp,%rdx)
       addq  $1, %rax
       cmpl  %eax, %r11d
       ja    .L113
.L112:
```

```
	leal	-4(%rsi), %r9d
	movl	%r12d, %r8d
	xorl	%edx, %edx
	leal	-3(%rsi), %eax
	shrl	$2, %r9d
	subl	%esi, %r8d
	leal	-2(%rsi), %r10d
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl(%rbx,%rax), %eax
	movzbl48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
	cmpl	%r8d, %edx
	jbe	.L117
.L118:
	xorl	%eax, %eax
	jmp	.L128
	.p2align 4,,10
.L129:
	movl	%edx, %ecx
	movl	$1, %eax
	leaq	(%rbx,%rcx), %r12
	cmpl	(%r12), %ebp
	jne	.L114
	movl	44(%rsp), %esi
	testl	%esi, %esi
	jle	.L124
	movl	(%r12,%r15), %esi
	cmpl	%esi, (%rdi,%r15)
	jne	.L114
	addq	%r14, %rcx
	xorl	%eax, %eax
	addq	%rbx, %rcx
	jmp	.L116
	.p2align 4,,10
.L132:
	movl	-5(%r13,%rax), %esi
	subq	$4, %rax
	cmpl	-1(%rcx,%rax), %esi
	jne	.L131
.L116:
	cmpq	%rax, %r9
	jne	.L132
.L124:
	movq	%r12, %rax
.L128:
	addq	$65592, %rsp
	popq	%rbx
	popq	%rsi
	popq	%rdi
	popq	%rbp
	popq	%r12
	popq	%r13
	popq	%r14
	popq	%r15
	ret
	.p2align 4,,10
.L131:
	movl	$1, %eax
	jmp	.L114
.L119:
	xorl	%r11d, %r11d
	jmp	.L112
*/

} //if (cbTarget<777)

		} else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
```

```
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv           PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                   PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                 PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                     PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                   PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                   PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                       PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                       PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                 PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                     PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
	FoundAtPosition = cbPattern - ((4)-1) + 1;
	PRIMALpositionCANDIDATE=i;
	while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
		j = PRIMALpositionCANDIDATE + 1;
		while ( j <= (FoundAtPosition-1) ) {
			if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
			j++;
		}
		PRIMALpositionCANDIDATE++;
	}
	PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
	if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
	if (cbPattern-i+1 <= PRIMALlength) break;
	if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv           PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                   PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                 PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                     PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                   PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                        PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                       PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                     PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                    PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                      00000000011111111112222222222333
                      12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
```

```
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for ‘12’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for ‘23’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for ‘zx’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for ‘xc’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for ‘23’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for ‘34’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for ‘zx’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for ‘xc’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for ‘xc’, FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = ‘1234567890qwertyuiopasdfghjklzxcv’



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for ‘vv’, FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = ‘vv’



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for ‘vv’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for ‘vB’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for ‘BO’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for ‘OO’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for ‘OM’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for ‘MS’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for ‘SH’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for ‘HA’, FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvvv  ! Searching for ‘AK’, FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvvv  ! Searching for ‘KA’, FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvvv  ! Searching for ‘AL’, FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvvv  ! Searching for ‘LA’, FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for ‘vv’, FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = ‘vvBOOMSHAKALA’
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length ‘PRIMALlength’ and compare the pbPattern[i] with length ‘cbPattern’;
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; ‘memset’ if not optimized
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
```

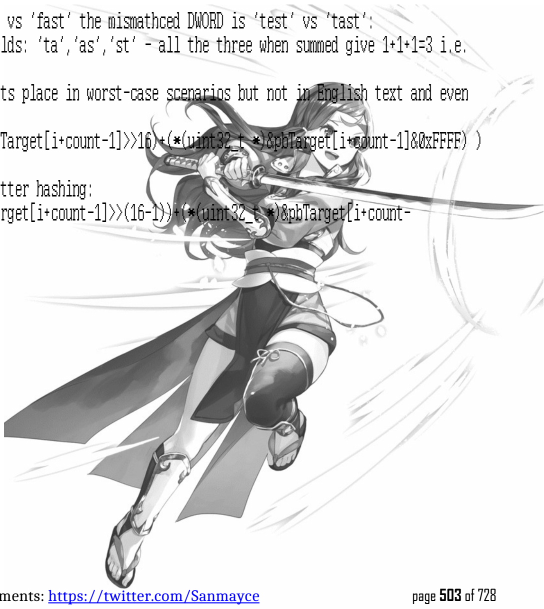


```
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // ‘Gulliver’ is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
```

```
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
}
*/
// Revision 2 commented section ]

    if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00    mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;

  if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
  } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                        if ( count <= 0 ) return(pbTarget+i);
  }
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);

    } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

        // BMH pseudo-order 4, needle should be >=8+2:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
  example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
        //"st f"
        //"t fo"
        //" fox"
        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
```

```
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
    #1
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
    the middle DWORD is 'bc'.
                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // Above line is replaced by next one with better hashing:
                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
    for ASCII texts with no symbols in range 128-255.
                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                // T           = 01010100
                                // TT          = 01010100 01010100
                                // TTTT        = 01010100 01010100 01010100 01010100
                                // TTTT>>16    = 00000000 00000000 01010100 01010100
                                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
    0 for English texts & 'ACGT' data.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // 'Maximus' uses branched 'if', again.
                                if ( \
                                ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
    1 )] ) == 0 \
                                || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
    1 )] ) == 0 \
                                ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//     DWORD #3                                                                       |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                     |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//     DWORD #3                                                                       |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                     |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                        // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
```

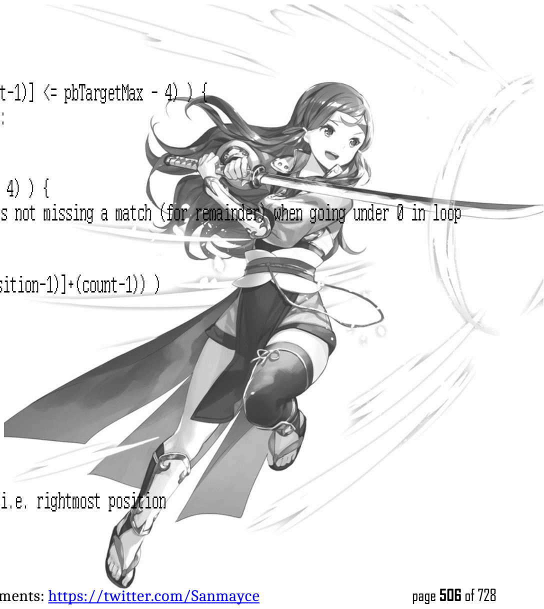
```
                                                count = cbPattern-4+1;
                                                // Below comparison is UNIdirectional:
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }

                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
    as it should
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
    Gulliver remains 1.
                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
    not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}
```



```
char * Railgun_Trolldom_a (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
       char * pbTargetMax = pbTarget + cbTarget;
       uint32_t ulHashPattern;
       signed long count;

       unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
       unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
       uint32_t i, Gulliver;

       uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
       uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
       uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

       if (cbPattern > cbTarget) return(NULL);

       if ( cbPattern<4 ) {
             // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
             // ...
             pbTarget = pbTarget+cbPattern;
             ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
             if ( cbPattern==3 ) {
                           for ( ;; ) {
                                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                     if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                        }
                                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                     pbTarget++;
                                                     if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                        }
                                        pbTarget++;
                                        if (pbTarget > pbTargetMax) return(NULL);
                           }
             } else {
             }
             for ( ;; ) {
                           if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                           if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                           pbTarget++;
                           if (pbTarget > pbTargetMax) return(NULL);
             }
       } else { //if ( cbPattern<4 )
             if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
            if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
```

```
                count--;
            }
            if ( count == 0) return((pbTarget-cbPattern));
           } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0¦1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0¦1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//     } else { //if ( cbPattern<4 )
//           if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                        //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] ¦
       (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                    if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
       == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                                        count = cbPattern-4+1;
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                    count = count-4;
                                                                        if ( count <= 0 ) return(pbTarget+i);
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1);
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
//           } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                                        count = cbPattern-4+1;
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                    count = count-4;
                                                                        if ( count <= 0 ) return(pbTarget+i);
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1);
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);

// Slower than Swampshine's simple 0¦1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
```

```
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
        cbPattern = PRIMALlengthCANDIDATE;
        pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                           // BMH order 2, needle should be >=4:
                           ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                           for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                           for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                           i=0;
                           while (i <= cbTarget-cbPattern) {
                                        Gulliver = 1; // 'Gulliver' is the skip
                                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                                      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                                   if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                                                                                count = cbPattern-4+1;
                                                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                              count = count-4;
                                                                                if ( count <= 0 ) return(pbTarget+i);
                                                                   }
                                                      }
                                        } else Gulliver = cbPattern-(2-1);
                                        i = i + Gulliver;
                                        //GlobalI++; // Comment it, it is only for stats.
                           }
                           return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                           ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                           for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                           // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                           // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                           // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                           // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                           //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                           for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                           i=0;
                           while (i <= cbTarget-cbPattern) {
                                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
        (Order-1) chars i.e. one char for Order 2
                                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                      if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                   count = cbPattern-4+1;
                                                                   while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
        // The line below is BUGGY:
        //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
        // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
        //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
        if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
              if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                           count = PRIMALlengthCANDIDATE-4+1;
                           while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                        count = count-4;
                           if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
              }
        }
}
                                                      }
                                                      Gulliver = 1;
                                        } else
                                                      Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                        }
                                        i = i + Gulliver;
                                        //GlobalI++; // Comment it, it is only for stats.
```



```
                    }
                    return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
--------------------------------------------------------------------------------------------
¦ testfile\Searcher                       ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
|---------------------------------------------------------------------------|-----------------------|
¦ Compiler                                ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
|---------------------------------------------------------------------------|-----------------------|
¦ The_Project_Gutenberg_EBook_of_Don      ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                 ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
|---------------------------------------------------------------------------|-----------------------|
¦ The_Project_Gutenberg_EBook_of_Dokoe    ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦          ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
|---------------------------------------------------------------------------|-----------------------|
¦ Dragonfly_genome_shotgun_sequence       ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                  ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
|---------------------------------------------------------------------------|-----------------------|
¦ LAOTZU_Wu_Wei_(BINARY).pdf              ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
|---------------------------------------------------------------------------|-----------------------|
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                    ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                    for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                    // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
                    //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                    // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00              mov r8d, 65536
//  0012d 44 8b 26        mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                    // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                    // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
     it is okay to use 'memset'. !
                    for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                    i=0;
                    while (i <= cbTarget-cbPattern) {
                              Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                              if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
     (Order-1) chars i.e. one char for Order 2
                              if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                  count = cbPattern-4+1;
                                                  while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                            count = count-4;
                                                  if ( count <= 0 ) return(pbTarget+i);
                                        }
                                        Gulliver = 1;
                              } else
                                        Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                              }
                              i = i + Gulliver;
                              //GlobalI++; // Comment it, it is only for stats.
                    }
                    return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
```

```
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
```

```
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
       pushq %r15
       .seh_pushreg      %r15
       movl  $65592, %eax
       pushq %r14
       .seh_pushreg      %r14
       pushq %r13
       .seh_pushreg      %r13
       pushq %r12
       .seh_pushreg      %r12
       pushq %rbp
       .seh_pushreg      %rbp
       pushq %rdi
       .seh_pushreg      %rdi
       pushq %rsi
       .seh_pushreg      %rsi
       pushq %rbx
       .seh_pushreg      %rbx
       call  ___chkstk_ms
       subq  %rax, %rsp
       .seh_stackalloc    65592
       .seh_endprologue
       cmpl  %r9d, %r8d
       movq  %rcx, %rbx
       movq  %rdx, %rdi
       movl  %r8d, %r12d
       movl  %r9d, %esi
       jb    .L118
       movl  (%rdx), %ebp
       leal  -1(%r9), %edx
```

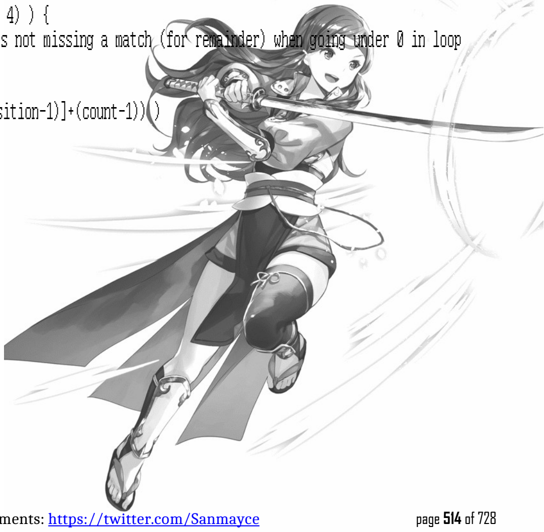
```
        movl  $65536, %r8d
        leaq  48(%rsp), %rcx
        movzbl%dl, %edx
        call  memset
        movl  %esi, %r11d
        subl  $1, %r11d
        je    .L119
        xorl  %eax, %eax
        .p2align 4,,10
.L113:
        movzwl(%rdi,%rax), %edx
        movb  %al, 48(%rsp,%rdx)
        addq  $1, %rax
        cmpl  %eax, %r11d
        ja    .L113
.L112:
        leal  -4(%rsi), %r9d
        movl  %r12d, %r8d
        xorl  %edx, %edx
        leal  -3(%rsi), %eax
        shrl  $2, %r9d
        subl  %esi, %r8d
        leal  -2(%rsi), %r10d
        movslq%eax, %r14
        negq  %r9
        movl  %eax, 44(%rsp)
        leaq  -1(%r14), %r15
        salq  $2, %r9
        leaq  (%rdi,%r14), %r13
        jmp   .L117
        .p2align 4,,10
.L130:
        movl  %r10d, %eax
        subl  %ecx, %eax
        cmpl  %r10d, %ecx
        je    .L129
.L114:
        addl  %eax, %edx
        cmpl  %r8d, %edx
        ja    .L118
.L117:
        leal  (%rdx,%r10), %eax
        movzwl(%rbx,%rax), %eax
        movzbl48(%rsp,%rax), %ecx
        cmpl  %r11d, %ecx
        jne   .L130
        movl  %r11d, %eax
        addl  %eax, %edx
        cmpl  %r8d, %edx
        jbe   .L117
.L118:
        xorl  %eax, %eax
        jmp   .L128
        .p2align 4,,10
.L129:
        movl  %edx, %ecx
        movl  $1, %eax
        leaq  (%rbx,%rcx), %r12
        cmpl  (%r12), %ebp
        jne   .L114
        movl  44(%rsp), %esi
        testl %esi, %esi
        jle   .L124
        movl  (%r12,%r15), %esi
        cmpl  %esi, (%rdi,%r15)
        jne   .L114
        addq  %r14, %rcx
        xorl  %eax, %eax
        addq  %rbx, %rcx
        jmp   .L116
        .p2align 4,,10
.L132:
        movl  -5(%r13,%rax), %esi
        subq  $4, %rax
        cmpl  -1(%rcx,%rax), %esi
        jne   .L131
.L116:
        cmpq  %rax, %r9
        jne   .L132
.L124:
        movq  %r12, %rax
.L128:
        addq  $65592, %rsp
        popq  %rbx
        popq  %rsi
        popq  %rdi
        popq  %rbp
        popq  %r12
        popq  %r13
```

```
        popq  %r14
        popq  %r15
        ret
        .p2align 4,,10
.L131:
        movl  $1, %eax
        jmp   .L114
.L119:
        xorl  %r11d, %r11d
        jmp   .L112
*/

} //if (cbTarget<777)

              } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
        while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
              j = PRIMALpositionCANDIDATE + 1;
              while ( j <= (FoundAtPosition-1) ) {
                            if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                            j++;
              }
              PRIMALpositionCANDIDATE++;
        }
        PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
        if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
        if (cbPattern-i+1 <= PRIMALlength) break;
        if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                    PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                  PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                      PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                    PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                         PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                        PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                      PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                      PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                     PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
```

```
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       000000000111111111122222222223333
                       123456789012345678901234567890123
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
```

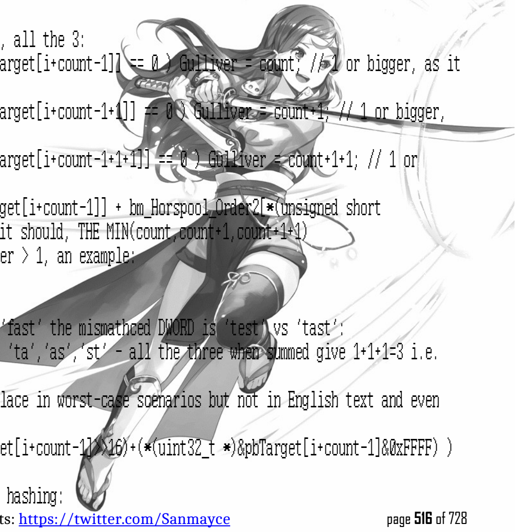
```
    Result:
    PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
```

```
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//       00                 mov r8d, 65536
//  00452 44 89 5c 24 20    mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60    mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff    call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
```



```
            } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0))>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0))>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                    Gulliver = 1;
                    //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                    // Above line is replaced by next one with better hashing:
                    if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                        // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then the middle DWORD is 'bc'.
                        // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                        // Above line is replaced by next one with better hashing:
                        // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy for ASCII texts with no symbols in range 128-255.
                        // Example for genomesque pair TT+TT being shifted by (16-1):
                        // T            = 01010100
                        // TT           = 01010100 01010100
                        // TTTT         = 01010100 01010100 01010100 01010100
                        // TTTT>>16     = 00000000 00000000 01010100 01010100
                        // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are 0 for English texts & 'ACGT' data.
                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                        // 'Maximus' uses branched 'if', again.
                        if ( \
                        ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-1 )] ) == 0 \
                        || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-1 )] ) == 0 \
                        ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                        // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                         |
//     DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!    |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = -------------------------  |
//         DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!       |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ----------------------------  |
//             DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                  |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = -------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                         |
//     DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!    |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = -------------------------  |
//         DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!       |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ----------------------------  |
//             DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                  |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = -------------------------------
                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
```

```
//              DWORD #1
                                        //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4])>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
     + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8])>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                  if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            // Order 4 [
                                                  // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                  // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
     'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                  //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                  //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                  //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                  //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                  //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                  //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                  //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                  //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                            count = cbPattern-4+1;
                                                            // Below comparison is UNIdirectional:
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                      count = count-4;

     if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
     // The line below is BUGGY:
     //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
     // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
     //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
     if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
          if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
     below:
                    count = PRIMALlengthCANDIDATE-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                              count = count-4;
                    if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
          }
     }
}
     } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                            if ( count <= 0 ) return(pbTarget+i);
     }


                                                            // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                            // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                                          for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                                    if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                                    if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
     = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
     *)&pbTarget[i] == ulHashPattern)
//                                                          }
//                                                          if ( count <= 0 ) return(pbTarget+i);
                                                                      // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                                      //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
     should
                                                                      //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
     as it should
                                                                      //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
     bigger, as it should
                                                            //        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
     *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                                      // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                                      // Needle:    fastest tax
                                                                      // Window: ...fastast tax...
                                                                      // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                                      // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
     Gulliver remains 1.
                                                                      // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
     not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                                      //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
     & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                                      // Above line is replaced by next one with better hashing:
```



```
```

```
//                                                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                 // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_b (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;

    uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
    uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
    uint32_t j, FoundAtPosition;

// Quadruplet [
  //char * pbTargetMax = pbTarget + cbTarget;
  //register unsigned long  ulHashPattern;
  uint32_t ulHashTarget;
  //unsigned long count;
  uint32_t countSTATIC;
  unsigned char SINGLET;
  uint32_t Quadruplet2nd;
  uint32_t Quadruplet3rd;
  uint32_t Quadruplet4th;
  uint32_t AdvanceHopperGrass;
// Quadruplet ]

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {
        // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
        // ...
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                        }
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
                }
        } else {
        }
        for ( ;; ) {
                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                pbTarget++;
                if (pbTarget > pbTargetMax) return(NULL);
        }
    } else { //if ( cbPattern<4 )
        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
     pbTarget = pbTarget+cbPattern;
     ulHashPattern = *(uint32_t *)(pbPattern);
//      countSTATIC = cbPattern-1;

  //SINGLET = *(char *)(pbPattern);
  SINGLET = ulHashPattern & 0xFF;
  Quadruplet2nd = SINGLET<<8;
  Quadruplet3rd = SINGLET<<16;
```

```
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//          }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//     } else { //if ( cbPattern<4 )
//           if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                        //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
       (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                    if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-2])&0x7)) )
       == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                                        count = cbPattern-4+1;
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                    count = count-4;
                                                                        if ( count <= 0 ) return(pbTarget+i);
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1);
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
//           } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                                        count = cbPattern-4+1;
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                    count = count-4;
                                                                        if ( count <= 0 ) return(pbTarget+i);
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1);
                                    i = i + Gulliver;
```

```
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
       cbPattern = PRIMALlengthCANDIDATE;
       pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
       (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
       // The line below is BUGGY:
       //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
       // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
       //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
       if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
```

```
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                     ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
|---------------------------------------------------------------------------------------|-----------------------|
¦ Compiler                              ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
|---------------------------------------------------------------------------------------|-----------------------|
¦ The_Project_Gutenberg_EBook_of_Don    ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt               ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                       ¦            ¦          ¦            ¦          ¦            ¦          ¦
|---------------------------------------------------------------------------------------|-----------------------|
¦ The_Project_Gutenberg_EBook_of_Dokoe  ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦        ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
|---------------------------------------------------------------------------------------|-----------------------|
¦ Dragonfly_genome_shotgun_sequence     ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                       ¦            ¦          ¦            ¦          ¦            ¦          ¦
|---------------------------------------------------------------------------------------|-----------------------|
¦ LAOTZU_Wu_Wei_(BINARY).pdf            ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                         ¦            ¦          ¦            ¦          ¦            ¦          ¦
|---------------------------------------------------------------------------------------|-----------------------|
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00              mov r8d, 65536
//  0012d 44 8b 26        mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
    it is okay to use 'memset'. !
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
                                        if ( count <= 0 ) return(pbTarget+i);
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
```

```
                                        }
                                        i = i + Gulliver;
                                        //GlobalI++; // Comment it, it is only for stats.
                           }
                           return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
```

```
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
       pushq %r15
       .seh_pushreg      %r15
       movl  $65592, %eax
       pushq %r14
       .seh_pushreg      %r14
       pushq %r13
       .seh_pushreg      %r13
       pushq %r12
       .seh_pushreg      %r12
       pushq %rbp
       .seh_pushreg      %rbp
       pushq %rdi
       .seh_pushreg      %rdi
```

```
	pushq	%rsi
	.seh_pushreg	%rsi
	pushq	%rbx
	.seh_pushreg	%rbx
	call	___chkstk_ms
	subq	%rax, %rsp
	.seh_stackalloc	65592
	.seh_endprologue
	cmpl	%r9d, %r8d
	movq	%rcx, %rbx
	movq	%rdx, %rdi
	movl	%r8d, %r12d
	movl	%r9d, %esi
	jb	.L118
	movl	(%rdx), %ebp
	leal	-1(%r9), %edx
	movl	$65536, %r8d
	leaq	48(%rsp), %rcx
	movzbl%dl, %edx
	call	memset
	movl	%esi, %r11d
	subl	$1, %r11d
	je	.L119
	xorl	%eax, %eax
	.p2align 4,,10
.L113:
	movzwl(%rdi,%rax), %edx
	movb	%al, 48(%rsp,%rdx)
	addq	$1, %rax
	cmpl	%eax, %r11d
	ja	.L113
.L112:
	leal	-4(%rsi), %r9d
	movl	%r12d, %r8d
	xorl	%edx, %edx
	leal	-3(%rsi), %eax
	shrl	$2, %r9d
	subl	%esi, %r8d
	leal	-2(%rsi), %r10d
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl(%rbx,%rax), %eax
	movzbl48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
	cmpl	%r8d, %edx
	jbe	.L117
.L118:
	xorl	%eax, %eax
	jmp	.L128
	.p2align 4,,10
.L129:
	movl	%edx, %ecx
	movl	$1, %eax
	leaq	(%rbx,%rcx), %r12
	cmpl	(%r12), %ebp
	jne	.L114
	movl	44(%rsp), %esi
	testl	%esi, %esi
	jle	.L124
	movl	(%r12,%r15), %esi
	cmpl	%esi, (%rdi,%r15)
	jne	.L114
	addq	%r14, %rcx
	xorl	%eax, %eax
	addq	%rbx, %rcx
	jmp	.L116
	.p2align 4,,10
.L132:
	movl	-5(%r13,%rax), %esi
```

```
    subq    $4, %rax
    cmpl    -1(%rcx,%rax), %esi
    jne .L131
.L116:
    cmpq    %rax, %r9
    jne .L132
.L124:
    movq    %r12, %rax
.L128:
    addq    $65592, %rsp
    popq    %rbx
    popq    %rsi
    popq    %rdi
    popq    %rbp
    popq    %r12
    popq    %r13
    popq    %r14
    popq    %r15
    ret
    .p2align 4,,10
.L131:
    movl    $1, %eax
    jmp .L114
.L119:
    xorl    %r11d, %r11d
    jmp .L112
*/

} //if (cbTarget<777)

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
    FoundAtPosition = cbPattern - ((4)-1) + 1;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
                if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
    if (cbPattern-i+1 <= PRIMALlength) break;
    if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                    PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                  PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                      PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                    PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                         PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                        PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                      PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                      PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                     PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
```

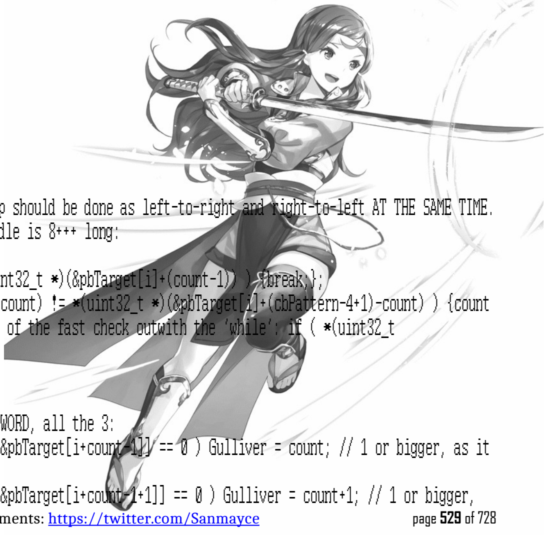
```
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       00000000011111111112222222222333
                       12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
```

```
Step 09_03: vvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
    Result:
    PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
      // The line below is BUGGY:
      //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
      // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
      //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
      if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
         if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
      below:
                        count = PRIMALlengthCANDIDATE-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                    count = count-4;
                        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
            }
      }
}
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
} else {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2+1)-2; else {
```

```
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00                    mov r8d, 65536
//  00452 44 89 5c 24 20        mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60        mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff        call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
```

```
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
  } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                        if ( count <= 0 ) return(pbTarget+i);
  }
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);

    } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

        // BMH pseudo-order 4, needle should be >=8+2:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
  example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
        //"fast"
        //"aste"
        //"stes"
        //"test"
        //"est "
        //"st f"
        //"t fo"
        //" fox"
        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
        //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0))>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        // Above line is replaced by next one with better hashing:
        for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0))>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1;
            //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
            // Above line is replaced by next one with better hashing:
            if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
  #1
                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
  0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
  the middle DWORD is 'bc'.
                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
  + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
  *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                // Above line is replaced by next one with better hashing:
                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
  for ASCII texts with no symbols in range 128-255.
                // Example for genomesque pair TT+TT being shifted by (16-1):
                // T           = 01010100
                // TT          = 01010100 01010100
                // TTTT        = 01010100 01010100 01010100 01010100
                // TTTT>>16    = 00000000 00000000 01010100 01010100
                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
  0 for English texts & 'ACGT' data.
                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
  1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
  *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                // 'Maximus' uses branched 'if', again.
                if ( \
                ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
  1 )] ) == 0 \
                || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
  1 )] ) == 0 \
                ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                      |
//    DWORD #3                                                                      |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!  |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ----------------------  |
//        DWORD #2                                                                     |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!     |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = -------------------------  |
//            DWORD #1                                                                    |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                               |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ----------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                      |
```

```
//     DWORD #3                                                   |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!     |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//            DWORD #1                                                |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                               |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                // Order 4 [
                                        // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                        // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
    'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                        //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                        //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                        //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                        //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                        //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                        //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                        //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                        //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                count = cbPattern-4+1;
                                                // Below comparison is UNIdirectional:
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }

                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
```



```
        as it should
                                                                                                  //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
        bigger, as it should
                                                                                     //           if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
        *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                                                                  // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                                                                  // Needle:    fastest tax
                                                                                                  // Window: ...fastast tax...
                                                                                                  // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                                                                  // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
        Gulliver remains 1.
                                                                                                  // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
        not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                                                                  //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
        & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                                                                  // Above line is replaced by next one with better hashing:
//                                                                                                if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
        1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                                                     // Order 4 ]
                                                                        }
                                                            }
                                                } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                                                i = i + Gulliver;
                                                //GlobalI++; // Comment it, it is only for stats.
                                    }
                                    return(NULL);

                        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
                        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_c (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
        char * pbTargetMax = pbTarget + cbTarget;
        uint32_t ulHashPattern;
        signed long count;

        unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
        unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
        uint32_t i, Gulliver;

        uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
        uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
        uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

        if (cbPattern > cbTarget) return(NULL);

        if ( cbPattern<4 ) {
                // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
                // ...
                pbTarget = pbTarget+cbPattern;
                ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
                if ( cbPattern==3 ) {
                                for ( ;; ) {
                                                if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                                }
                                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                                pbTarget++;
                                                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                                }
                                                pbTarget++;
                                                if (pbTarget > pbTargetMax) return(NULL);
                                }
                } else {
                }
                for ( ;; ) {
                                if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                                if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                                pbTarget++;
                                if (pbTarget > pbTargetMax) return(NULL);
                }
        } else { //if ( cbPattern<4 )
                if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {
```

```
// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//          count = countSTATIC;
//          while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//             if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//          }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//     } else { //if ( cbPattern<4 )
//           if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                        //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
       (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                    if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
       == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                                        count = cbPattern-4+1;
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                    count = count-4;
                                                                        if ( count <= 0 ) return(pbTarget+i);
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1);
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
//           } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
```

```
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
        cbPattern = PRIMALlengthCANDIDATE;
        pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
        (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
```

```
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
       // The line below is BUGGY:
       //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
       // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
       //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
       if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
             if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                          count = PRIMALlengthCANDIDATE-4+1;
                          while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                       count = count-4;
                          if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
             }
       }
}
                                                    }
                                                    Gulliver = 1;
                                       } else
                                                    Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                       }
                                       i = i + Gulliver;
                                       //GlobalI++; // Comment it, it is only for stats.
                          }
                          return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
-------------------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                         ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦---------------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                                  ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦---------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don        ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                   ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe      ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence         ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                    ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                           ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------------------¦-----------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf                ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                             ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦---------------------------------------------------------------------------------------------¦-----------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                          ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                          for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                          // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
                          //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                          // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00              mov r8d, 65536
//  0012d 44 8b 26        mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                          // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                          // we must switch to 0¦1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
       it is okay to use 'memset'. !
```

```
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
            if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
            if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                    count = cbPattern-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                        count = count-4;
                    if ( count <= 0 ) return(pbTarget+i);
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
```

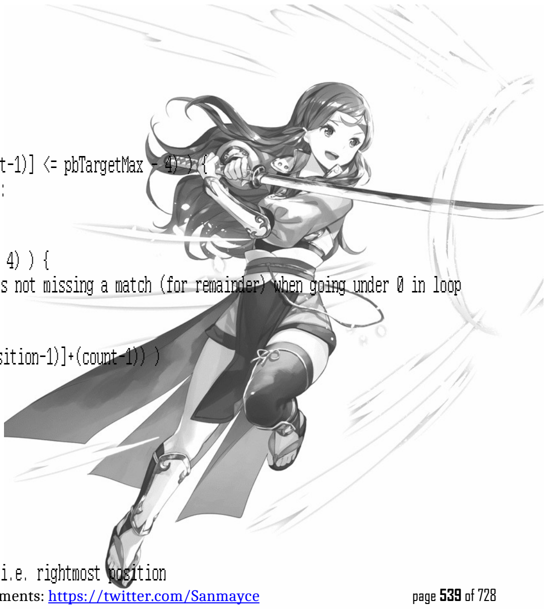
```
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/
```

```
// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
        pushq %r15
        .seh_pushreg    %r15
        movl  $65592, %eax
        pushq %r14
        .seh_pushreg    %r14
        pushq %r13
        .seh_pushreg    %r13
        pushq %r12
        .seh_pushreg    %r12
        pushq %rbp
        .seh_pushreg    %rbp
        pushq %rdi
        .seh_pushreg    %rdi
        pushq %rsi
        .seh_pushreg    %rsi
        pushq %rbx
        .seh_pushreg    %rbx
        call  ___chkstk_ms
        subq  %rax, %rsp
        .seh_stackalloc   65592
        .seh_endprologue
        cmpl  %r9d, %r8d
        movq  %rcx, %rbx
        movq  %rdx, %rdi
        movl  %r8d, %r12d
        movl  %r9d, %esi
        jb    .L118
        movl  (%rdx), %ebp
        leal  -1(%r9), %edx
        movl  $65536, %r8d
        leaq  48(%rsp), %rcx
        movzbl%dl, %edx
        call  memset
        movl  %esi, %r11d
        subl  $1, %r11d
        je    .L119
        xorl  %eax, %eax
        .p2align 4,,10
.L113:
        movzwl(%rdi,%rax), %edx
        movb  %al, 48(%rsp,%rdx)
        addq  $1, %rax
        cmpl  %eax, %r11d
        ja    .L113
.L112:
        leal  -4(%rsi), %r9d
        movl  %r12d, %r8d
        xorl  %edx, %edx
        leal  -3(%rsi), %eax
        shrl  $2, %r9d
        subl  %esi, %r8d
        leal  -2(%rsi), %r10d
        movslq%eax, %r14
        negq  %r9
        movl  %eax, 44(%rsp)
        leaq  -1(%r14), %r15
        salq  $2, %r9
        leaq  (%rdi,%r14), %r13
        jmp   .L117
        .p2align 4,,10
.L130:
        movl  %r10d, %eax
        subl  %ecx, %eax
        cmpl  %r10d, %ecx
        je    .L129
.L114:
        addl  %eax, %edx
        cmpl  %r8d, %edx
        ja    .L118
.L117:
        leal  (%rdx,%r10), %eax
        movzwl(%rbx,%rax), %eax
        movzbl48(%rsp,%rax), %ecx
        cmpl  %r11d, %ecx
        jne   .L130
        movl  %r11d, %eax
        addl  %eax, %edx
        cmpl  %r8d, %edx
        jbe   .L117
.L118:
        xorl  %eax, %eax
        jmp   .L128
        .p2align 4,,10
.L129:
        movl  %edx, %ecx
        movl  $1, %eax
```

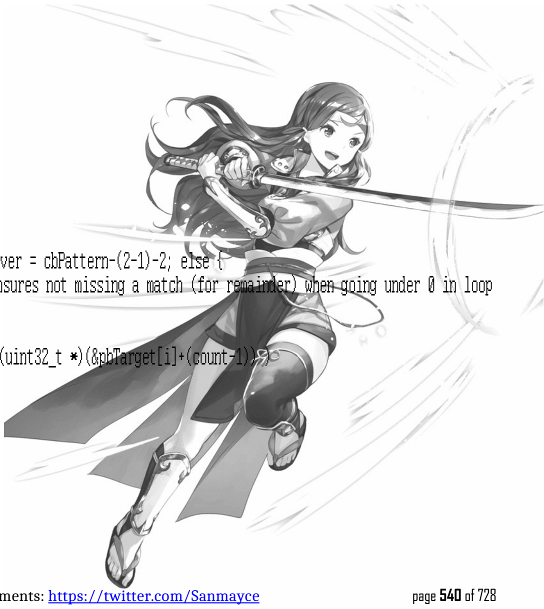
```
        leaq    (%rbx,%rcx), %r12
        cmpl    (%r12), %ebp
        jne     .L114
        movl    44(%rsp), %esi
        testl   %esi, %esi
        jle     .L124
        movl    (%r12,%r15), %esi
        cmpl    %esi, (%rdi,%r15)
        jne     .L114
        addq    %r14, %rcx
        xorl    %eax, %eax
        addq    %rbx, %rcx
        jmp     .L116
        .p2align 4,,10
.L132:
        movl    -5(%r13,%rax), %esi
        subq    $4, %rax
        cmpl    -1(%rcx,%rax), %esi
        jne     .L131
.L116:
        cmpq    %rax, %r9
        jne     .L132
.L124:
        movq    %r12, %rax
.L128:
        addq    $65592, %rsp
        popq    %rbx
        popq    %rsi
        popq    %rdi
        popq    %rbp
        popq    %r12
        popq    %r13
        popq    %r14
        popq    %r15
        ret
        .p2align 4,,10
.L131:
        movl    $1, %eax
        jmp     .L114
.L119:
        xorl    %r11d, %r11d
        jmp     .L112
*/

} //if (cbTarget<777)

                } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
        while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
                j = PRIMALpositionCANDIDATE + 1;
                while ( j <= (FoundAtPosition-1) ) {
                                if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                                j++;
                }
                PRIMALpositionCANDIDATE++;
        }
        PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
        if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
        if (cbPattern-i+1 <= PRIMALlength) break;
        if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv             PRIMALposition=31 PRIMALlength=02  'vv'
```

```
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                   PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                 PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                     PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                   PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                        PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                       PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                     PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                    PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       000000000111111111122222222223333
                       123456789012345678901234567890123
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'
```

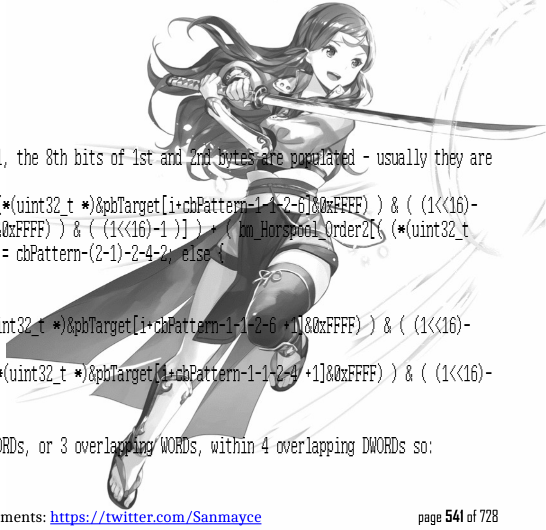
```
    PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
    Result:
    PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                    ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                    for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                    for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                    i=0;
                    while (i <= cbTarget-cbPattern) {
                              Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                              if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                              if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                  count = cbPattern-4+1;
                                                  while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                            count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
      // The line below is BUGGY:
      //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
      // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
      //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
      if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
          if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
      below:
                    count = PRIMALlengthCANDIDATE-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                              count = count-4;
                    if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
          }
      }
}
                                        }
                                        Gulliver = 1;
                              } else
                                        Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
```



```
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} else {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00            mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
```



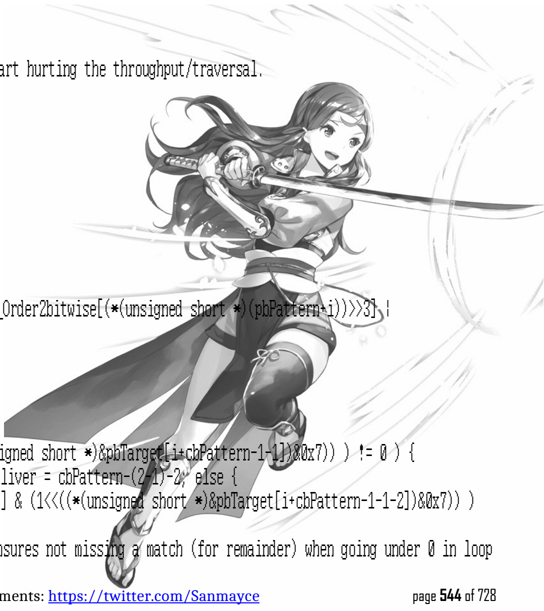
```
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
    #1
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
    the middle DWORD is 'bc'.
                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // Above line is replaced by next one with better hashing:
                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
    for ASCII texts with no symbols in range 128-255.
                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                // T           = 01010100
                                // TT          = 01010100 01010100
                                // TTTT        = 01010100 01010100 01010100 01010100
                                // TTTT>>16    = 00000000 00000000 01010100 01010100
                                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
    0 for English texts & 'ACGT' data.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
    1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // 'Maximus' uses branched 'if', again.
                                if ( \
                                ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
    1 )] ) == 0 \
                                || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
    1 )] ) == 0 \
                                ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
```



```
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6])>>16) =    !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//     DWORD #3                                                                       |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4])>>16) =    !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2])>>16) =    !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                     |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0])>>16) =                                |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//     DWORD #3                                                                       |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                     |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0])>>16) =                                |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                                                   //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
        1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
        *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
                                                   //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4])>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
        + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8])>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                               if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                           // Order 4 [
                                                               // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                               // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
        'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                               //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                               //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                               //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                               //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                               //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                               //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                               //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                               //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                                           count = cbPattern-4+1;
                                                                           // Below comparison is UNIdirectional:
                                                                           while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                       count = count-4;

        if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
        // The line below is BUGGY:
        //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
        // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
        //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
        if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
              if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
        below:
                          count = PRIMALlengthCANDIDATE-4+1;
                          while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                      count = count-4;
                          if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
              }
        }
}
        } else { //if (cbPattern != PRIMALlengthCANDIDATE)
```

```
                                                if ( count <= 0 ) return(pbTarget+i);
    }

                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
    as it should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
    Gulliver remains 1.
                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
    not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_d (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;

    uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
    uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
    uint32_t j, FoundAtPosition;

// Quadruplet [
  //char * pbTargetMax = pbTarget + cbTarget;
  //register unsigned long  ulHashPattern;
  uint32_t ulHashTarget;
  //unsigned long count;
  uint32_t countSTATIC;
  unsigned char SINGLET;
  uint32_t Quadruplet2nd;
  uint32_t Quadruplet3rd;
  uint32_t Quadruplet4th;
  uint32_t AdvanceHopperGrass;
// Quadruplet ]

    if (cbPattern > cbTarget) return(NULL);

    if ( cbPattern<4 ) {
        // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
        // ...
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
        if ( cbPattern==3 ) {
                for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                        }
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                pbTarget++;
```

```
                                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                    }
                                    pbTarget++;
                                    if (pbTarget > pbTargetMax) return(NULL);
                        }
            } else {
            }
            for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
            }
      } else { //if ( cbPattern<4 )
            if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
       pbTarget = pbTarget+cbPattern;
       ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

   //SINGLET = *(char *)(pbPattern);
   SINGLET = ulHashPattern & 0xFF;
   Quadruplet2nd = SINGLET<<8;
   Quadruplet3rd = SINGLET<<16;
   Quadruplet4th = SINGLET<<24;

   for ( ;; )
   {
      AdvanceHopperGrass = 0;
      ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

       if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//              count--;
//         }
        count = cbPattern-1;
        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
             if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
              count--;
        }
        if ( count == 0) return((pbTarget-cbPattern));
       } else { // The goal here: to avoid memory accesses by stressing the registers.
   if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
        AdvanceHopperGrass++;
        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
             AdvanceHopperGrass++;
             if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
        }
   }
      }

      AdvanceHopperGrass++;

      pbTarget = pbTarget + AdvanceHopperGrass;
       if (pbTarget > pbTargetMax)
           return(NULL);
   }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//      } else { //if ( cbPattern<4 )
//            if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                        //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
      (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                    if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
      == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
      below:
```



```
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
//      } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

// Slower than Swampshine's simple 0:1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
       cbPattern = PRIMALlengthCANDIDATE;
       pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
       below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
```

```
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
---------------------------------------------------------------------------------------------------
| testfile\Searcher                     | GNU/GLIBC memmem()    | Railgun_Swampshine    | Railgun_Trolldom      |
|---------------------------------------------------------------------------|-----------------------|
| Compiler                              | Intel 15.0 | GCC 5.10 | Intel 15.0 | GCC 5.10 | Intel 15.0 | GCC 5.10 |
|---------------------------------------------------------------------------|-----------------------|
| The_Project_Gutenberg_EBook_of_Don    |        190 |      226 |       1654 |     1729 |       1147 |     1764 |
| _Quixote_996_(ANSI).txt               |            |          |            |          |            |          |
| 2,347,772 bytes                       |            |          |            |          |            |          |
|---------------------------------------------------------------------------|-----------------------|
| The_Project_Gutenberg_EBook_of_Dokoe  |        582 |      760 |       3094 |     2898 |       2410 |     3036 |
| _by_Hakucho_Masamune_(Japanese_UTF-8).txt |        |          |            |          |            |          |
| 899,425 bytes                         |            |          |            |          |            |          |
|---------------------------------------------------------------------------|-----------------------|
| Dragonfly_genome_shotgun_sequence     |        104 |      109 |        445 |      458 |        484 |      553 |
| _(ACGT_alphabet).fasta                |            |          |            |          |            |          |
| 4,487,433 bytes                       |            |          |            |          |            |          |
|---------------------------------------------------------------------------|-----------------------|
| LAOTZU_Wu_Wei_(BINARY).pdf            |         99 |      144 |        629 |      580 |        185 |      570 |
| 954,035 bytes                         |            |          |            |          |            |          |
|---------------------------------------------------------------------------|-----------------------|
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
```

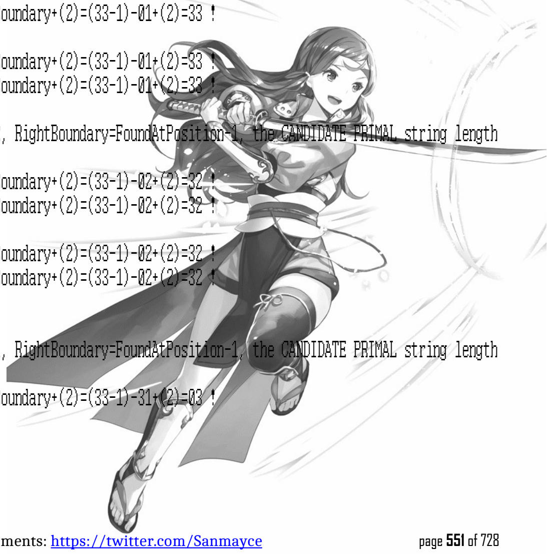
```
//       70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//       80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//       90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2          mov edx, eax
//  00167 72 d6          jb .B5.21
                         //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                         // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//       00              mov r8d, 65536
//  0012d 44 8b 26       mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                         // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                         // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
        it is okay to use 'memset'. !
                         for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                         i=0;
                         while (i <= cbTarget-cbPattern) {
                                     Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                     if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
        (Order-1) chars i.e. one char for Order 2
                                     if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                 if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                             count = cbPattern-4+1;
                                                             while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                         count = count-4;
                                                             if ( count <= 0 ) return(pbTarget+i);
                                                 }
                                                 Gulliver = 1;
                                     } else
                                                 Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                     }
                                     i = i + Gulliver;
                                     //GlobalI++; // Comment it, it is only for stats.
                         }
                         return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
```

```
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
```

```
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
	pushq	%r15
	.seh_pushreg	%r15
	movl	$65592, %eax
	pushq	%r14
	.seh_pushreg	%r14
	pushq	%r13
	.seh_pushreg	%r13
	pushq	%r12
	.seh_pushreg	%r12
	pushq	%rbp
	.seh_pushreg	%rbp
	pushq	%rdi
	.seh_pushreg	%rdi
	pushq	%rsi
	.seh_pushreg	%rsi
	pushq	%rbx
	.seh_pushreg	%rbx
	call	___chkstk_ms
	subq	%rax, %rsp
	.seh_stackalloc	65592
	.seh_endprologue
	cmpl	%r9d, %r8d
	movq	%rcx, %rbx
	movq	%rdx, %rdi
	movl	%r8d, %r12d
	movl	%r9d, %esi
	jb	.L118
	movl	(%rdx), %ebp
	leal	-1(%r9), %edx
	movl	$65536, %r8d
	leaq	48(%rsp), %rcx
	movzbl%dl, %edx
	call	memset
	movl	%esi, %r11d
	subl	$1, %r11d
	je	.L119
	xorl	%eax, %eax
	.p2align 4,,10
.L113:
	movzwl(%rdi,%rax), %edx
	movb	%al, 48(%rsp,%rdx)
	addq	$1, %rax
	cmpl	%eax, %r11d
	ja	.L113
.L112:
	leal	-4(%rsi), %r9d
	movl	%r12d, %r8d
	xorl	%edx, %edx
	leal	-3(%rsi), %eax
	shrl	$2, %r9d
	subl	%esi, %r8d
	leal	-2(%rsi), %r10d
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
```

```
        leal  (%rdx,%r10), %eax
        movzwl(%rbx,%rax), %eax
        movzbl48(%rsp,%rax), %ecx
        cmpl  %r11d, %ecx
        jne   .L130
        movl  %r11d, %eax
        addl  %eax, %edx
        cmpl  %r8d, %edx
        jbe   .L117
.L118:
        xorl  %eax, %eax
        jmp   .L128
        .p2align 4,,10
.L129:
        movl  %edx, %ecx
        movl  $1, %eax
        leaq  (%rbx,%rcx), %r12
        cmpl  (%r12), %ebp
        jne   .L114
        movl  44(%rsp), %esi
        testl %esi, %esi
        jle   .L124
        movl  (%r12,%r15), %esi
        cmpl  %esi, (%rdi,%r15)
        jne   .L114
        addq  %r14, %rcx
        xorl  %eax, %eax
        addq  %rbx, %rcx
        jmp   .L116
        .p2align 4,,10
.L132:
        movl  -5(%r13,%rax), %esi
        subq  $4, %rax
        cmpl  -1(%rcx,%rax), %esi
        jne   .L131
.L116:
        cmpq  %rax, %r9
        jne   .L132
.L124:
        movq  %r12, %rax
.L128:
        addq  $65592, %rsp
        popq  %rbx
        popq  %rsi
        popq  %rdi
        popq  %rbp
        popq  %r12
        popq  %r13
        popq  %r14
        popq  %r15
        ret
        .p2align 4,,10
.L131:
        movl  $1, %eax
        jmp   .L114
.L119:
        xorl  %r11d, %r11d
        jmp   .L112
*/

} //if (cbTarget<777)

                } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                   PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                 PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                     PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                   PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                   PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                       PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                       PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                 PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                     PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
        while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
                j = PRIMALpositionCANDIDATE + 1;
```

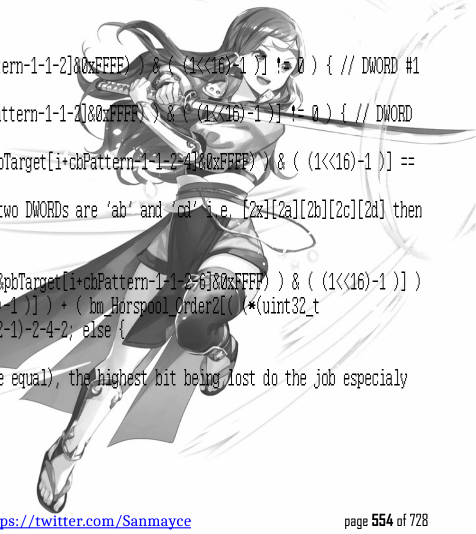
```
            while ( j <= (FoundAtPosition-1) ) {
                        if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                        j++;
            }
            PRIMALpositionCANDIDATE++;
       }
       PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
       if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
       if (cbPattern-i+1 <= PRIMALlength) break;
       if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv           PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv           PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv           PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                   PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                 PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                             PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                     PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                    PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                   PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                   PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                        PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                       PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                    PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                     PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                    PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       000000000111111111122222222223333
                       123456789012345678901234567890123
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
```



```
     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvBOOMSHAKALAKAvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvBOOMSHAKALAKAvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
       (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
       // The line below is BUGGY:
       //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
       // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
```

```
      //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
      if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
            if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
      below:
                        count = PRIMALlengthCANDIDATE-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                    count = count-4;
                        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
            }
      }
}
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
} else {
                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = 1; // 'Gulliver' is the skip
                                    if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                                            if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
      below:
                                                                        count = cbPattern-4+1;
                                                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                    count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
      // The line below is BUGGY:
      //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
      // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
      //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
      if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
            if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
      below:
                        count = PRIMALlengthCANDIDATE-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                                    count = count-4;
                        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
            }
      }
}
                                                            }
                                                }
                                    } else Gulliver = cbPattern-(2-1);
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
}
*/
// Revision 2 commented section ]

            if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                        // BMH order 2, needle should be >=4:
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//                      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                        // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00            mov r8d, 65536
//  00452 44 89 5c 24 20   mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60   mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff   call _intel_fast_memset
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                        i=0;
                        while (i <= cbTarget-cbPattern) {
```

```
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
    #1
                                //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
    0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
    the middle DWORD is 'bc'.
                                // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
    + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
    *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                // Above line is replaced by next one with better hashing:
                                // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
    for ASCII texts with no symbols in range 128-255.
                                // Example for genomesque pair TT+TT being shifted by (16-1):
                                // T            = 01010100
                                // TT           = 01010100 01010100
                                // TTTT         = 01010100 01010100 01010100 01010100
```



```
                // TTTT>>16     = 00000000 00000000 01010100 01010100
                // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are 0 for English texts & 'ACGT' data.
                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                // 'Maximus' uses branched 'if', again.
                if ( \
                ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-1 )] ) == 0 \
                || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-1 )] ) == 0 \
                ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =    !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//    DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =    !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//        DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =    !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//            DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                        |
//    DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//        DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//            DWORD #1                                                                      |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                 |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//        DWORD #2
//                DWORD #1
                //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                        // Order 4 [
                    // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                    // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                    //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                    //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                    //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                    //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                    //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                    //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                    //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                    //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                        count = cbPattern-4+1;
                        // Below comparison is UNIdirectional:
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
```

```
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }

                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,

    as it should

                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or

    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.

    Gulliver remains 1.
                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even

    not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )

    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_e (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;

    uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
    uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
    uint32_t j, FoundAtPosition;

// Quadruplet [
  //char * pbTargetMax = pbTarget + cbTarget;
  //register unsigned long  ulHashPattern;
  uint32_t ulHashTarget;
  //unsigned long count;
  uint32_t countSTATIC;
  unsigned char SINGLET;
  uint32_t Quadruplet2nd;
  uint32_t Quadruplet3rd;
  uint32_t Quadruplet4th;
  uint32_t AdvanceHopperGrass;
```

```
// Quadruplet ]

       if (cbPattern > cbTarget) return(NULL);

       if ( cbPattern<4 ) {
             // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
             // ...
             pbTarget = pbTarget+cbPattern;
             ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
             if ( cbPattern==3 ) {
                           for ( ;; ) {
                                        if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                      if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                        }
                                        if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                      pbTarget++;
                                                      if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                        }
                                        pbTarget++;
                                        if (pbTarget > pbTargetMax) return(NULL);
                           }
             } else {
             }
             for ( ;; ) {
                           if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                           if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                           pbTarget++;
                           if (pbTarget > pbTargetMax) return(NULL);
             }
       } else { //if ( cbPattern<4 )
             if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome ‘Railgun_Quadruplet’, it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case ‘if (cbPattern==1)’ was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than ‘Boyer_Moore_Horspool’.
{
        pbTarget = pbTarget+cbPattern;
        ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
       AdvanceHopperGrass = 0;
       ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

        if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but ‘AdvanceHopperGrass’ must be calculated - it has a higher priority.
//           count = countSTATIC;
//           while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//              if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//                count--;
//           }
         count = cbPattern-1;
         while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
              if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
               count--;
         }
         if ( count == 0) return((pbTarget-cbPattern));
        } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
         AdvanceHopperGrass++;
         if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
              AdvanceHopperGrass++;
              if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
         }
    }
       }

       AdvanceHopperGrass++;

       pbTarget = pbTarget + AdvanceHopperGrass;
        if (pbTarget > pbTargetMax)
            return(NULL);
    }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for ‘Don_Quixote’ with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for ‘Don_Quixote’ with GCC:
//     } else { //if ( cbPattern<4 )
//           if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                           // BMH order 2, needle should be >=4:
```

```
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
        //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
  (1<<((*(unsigned short *)(pbPattern+i))&0x7));
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
            if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-2])&0x7)) )
  == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
                        if ( count <= 0 ) return(pbTarget+i);
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
//  } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//      for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
                        if ( count <= 0 ) return(pbTarget+i);
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
       cbPattern = PRIMALlengthCANDIDATE;
       pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                        count = cbPattern-4+1;
```

```
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
                        if ( count <= 0 ) return(pbTarget+i);
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
        // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
        // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
        // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
            if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
  (Order-1) chars i.e. one char for Order 2
            if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                    count = cbPattern-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
---------------------------------------------------------------------------------------------------------------
¦ testfile\Searcher                          ¦ GNU/GLIBC memmem()     ¦ Railgun_Swampshine     ¦ Railgun_Trolldom       ¦
¦-------------------------------------------------------------------------------------¦------------------------¦
¦ Compiler                                   ¦ Intel 15.0 ¦ GCC 5.10  ¦ Intel 15.0 ¦ GCC 5.10  ¦ Intel 15.0 ¦ GCC 5.10  ¦
¦-------------------------------------------------------------------------------------¦------------------------¦
¦ The_Project_Gutenberg_EBook_of_Don         ¦        190 ¦       226 ¦       1654 ¦      1729 ¦       1147 ¦      1764 ¦
¦ _Quixote_996_(ANSI).txt                    ¦            ¦           ¦            ¦           ¦            ¦           ¦
¦ 2,347,772 bytes                            ¦            ¦           ¦            ¦           ¦            ¦           ¦
¦-------------------------------------------------------------------------------------¦------------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe       ¦        582 ¦       760 ¦       3094 ¦      2898 ¦       2410 ¦      3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt  ¦            ¦           ¦            ¦           ¦            ¦           ¦
¦ 899,425 bytes                              ¦            ¦           ¦            ¦           ¦            ¦           ¦
¦-------------------------------------------------------------------------------------¦------------------------¦
¦ Dragonfly_genome_shotgun_sequence          ¦        104 ¦       109 ¦        445 ¦       458 ¦        484 ¦       553 ¦
¦ _(ACGT_alphabet).fasta                     ¦            ¦           ¦            ¦           ¦            ¦           ¦
¦ 4,487,433 bytes                            ¦            ¦           ¦            ¦           ¦            ¦           ¦
¦-------------------------------------------------------------------------------------¦------------------------¦
```

```
¦ LAOTZU_Wu_Wei_(BINARY).pdf              ¦        99 ¦      144 ¦       629 ¦      580 ¦       185 ¦      570 ¦
¦ 954,035 bytes                           ¦           ¦          ¦           ¦          ¦           ¦          ¦
|-------------------------------------------------------------------------------------|-----------------------|
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40        add eax, 64
//  00142 66 0f 7f 44 14
//        60              movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00  cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70              movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00     movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00     movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2           mov edx, eax
//  00167 72 d6           jb .B5.21
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00              mov r8d, 65536
//  0012d 44 8b 26        mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff  call _intel_fast_memset
                        // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                        // we must switch to 0|1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 -
      it is okay to use 'memset'. !
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
      (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
                                                            if ( count <= 0 ) return(pbTarget+i);
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
      ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
```

```
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0               ;3492.33
        mov       r10d, eax                                    ;3492.4
        jb        .B6.4         ; Prob 99%                     ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                     ;3515.28
        je        .B6.12        ; Prob 50%                     ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                       ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                      ;3515.4
        mov       r15d, r11d                                   ;3515.4
        xor       r10d, r10d                                   ;3515.4
        shr       r15d, 1                                      ;3515.4
        test      r15d, r15d                                   ;3515.4
        jbe       .B6.10        ; Prob 15%                     ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                     ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                    ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                   ;3515.36
        inc       r10d                                         ;3515.4
        cmp       r10d, r15d                                   ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                    ;3515.36
        jb        .B6.8         ; Prob 64%                     ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                   ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                          ;3515.36
        cmp       eax, r11d                                    ;3515.4
        jae       .B6.12        ; Prob 15%                     ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                     ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                    ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                   ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                      ;3522.27
        movsxd    r15, r15d                                    ;3522.7
        sub       r8d, r9d                                     ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                   ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                      ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                   ;3518.16
        cmp       eax, ebp                                     ;3519.32
        je        .B6.24        ; Prob 50%                     ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                    ;3520.32
        jne       .B6.23        ; Prob 62%                     ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                    ;3521.25
        add       rax, rcx                                     ;3521.25
        cmp       r13d, DWORD PTR [rax]                        ;3521.40
        je        .B6.17        ; Prob 50%                     ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                       ;3527.6
        jmp       .B6.24        ; Prob 100%                    ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                     ;3522.7
        test      r15, r15                                     ;3523.23
        jle       .B6.22        ; Prob 2%                      ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                      ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                  ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                  ;3523.79
        jne       .B6.26        ; Prob 20%                     ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                      ;3524.22
        test      rbx, rbx                                     ;3523.23
        jg        .B6.19        ; Prob 82%                     ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                      ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
```

```
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
        pushq %r15
        .seh_pushreg      %r15
        movl  $65592, %eax
        pushq %r14
        .seh_pushreg      %r14
        pushq %r13
        .seh_pushreg      %r13
        pushq %r12
        .seh_pushreg      %r12
        pushq %rbp
        .seh_pushreg      %rbp
        pushq %rdi
        .seh_pushreg      %rdi
        pushq %rsi
        .seh_pushreg      %rsi
        pushq %rbx
        .seh_pushreg      %rbx
        call  ___chkstk_ms
        subq  %rax, %rsp
        .seh_stackalloc    65592
        .seh_endprologue
        cmpl  %r9d, %r8d
        movq  %rcx, %rbx
        movq  %rdx, %rdi
        movl  %r8d, %r12d
        movl  %r9d, %esi
        jb    .L118
        movl  (%rdx), %ebp
        leal  -1(%r9), %edx
        movl  $65536, %r8d
        leaq  48(%rsp), %rcx
        movzbl%dl, %edx
        call  memset
        movl  %esi, %r11d
        subl  $1, %r11d
        je    .L119
        xorl  %eax, %eax
        .p2align 4,,10
.L113:
        movzwl(%rdi,%rax), %edx
        movb  %al, 48(%rsp,%rdx)
        addq  $1, %rax
        cmpl  %eax, %r11d
        ja    .L113
.L112:
        leal  -4(%rsi), %r9d
        movl  %r12d, %r8d
        xorl  %edx, %edx
        leal  -3(%rsi), %eax
        shrl  $2, %r9d
        subl  %esi, %r8d
        leal  -2(%rsi), %r10d
        movslq%eax, %r14
        negq  %r9
```

```
        movl    %eax, 44(%rsp)
        leaq    -1(%r14), %r15
        salq    $2, %r9
        leaq    (%rdi,%r14), %r13
        jmp     .L117
        .p2align 4,,10
.L130:
        movl    %r10d, %eax
        subl    %ecx, %eax
        cmpl    %r10d, %ecx
        je      .L129
.L114:
        addl    %eax, %edx
        cmpl    %r8d, %edx
        ja      .L118
.L117:
        leal    (%rdx,%r10), %eax
        movzwl  (%rbx,%rax), %eax
        movzbl  48(%rsp,%rax), %ecx
        cmpl    %r11d, %ecx
        jne     .L130
        movl    %r11d, %eax
        addl    %eax, %edx
        cmpl    %r8d, %edx
        jbe     .L117
.L118:
        xorl    %eax, %eax
        jmp     .L128
        .p2align 4,,10
.L129:
        movl    %edx, %ecx
        movl    $1, %eax
        leaq    (%rbx,%rcx), %r12
        cmpl    (%r12), %ebp
        jne     .L114
        movl    44(%rsp), %esi
        testl   %esi, %esi
        jle     .L124
        movl    (%r12,%r15), %esi
        cmpl    %esi, (%rdi,%r15)
        jne     .L114
        addq    %r14, %rcx
        xorl    %eax, %eax
        addq    %rbx, %rcx
        jmp     .L116
        .p2align 4,,10
.L132:
        movl    -5(%r13,%rax), %esi
        subq    $4, %rax
        cmpl    -1(%rcx,%rax), %esi
        jne     .L131
.L116:
        cmpq    %rax, %r9
        jne     .L132
.L124:
        movq    %r12, %rax
.L128:
        addq    $65592, %rsp
        popq    %rbx
        popq    %rsi
        popq    %rdi
        popq    %rbp
        popq    %r12
        popq    %r13
        popq    %r14
        popq    %r15
        ret
        .p2align 4,,10
.L131:
        movl    $1, %eax
        jmp     .L114
.L119:
        xorl    %r11d, %r11d
        jmp     .L112
*/

} //if (cbTarget<777)

                } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv                PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv                 PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv                PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                        PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                      PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                                  PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                          PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                         PRIMALposition=01 PRIMALlength=08  'Sandokan'
```

```
// Needle: shazamish                                PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                    PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                    PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                              PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                  PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                             PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                             PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
       FoundAtPosition = cbPattern - ((4)-1) + 1;
       PRIMALpositionCANDIDATE=i;
       while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
             j = PRIMALpositionCANDIDATE + 1;
             while ( j <= (FoundAtPosition-1) ) {
                          if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                          j++;
             }
             PRIMALpositionCANDIDATE++;
       }
       PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
       if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
       if (cbPattern-i+1 <= PRIMALlength) break;
       if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                              PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                          PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                  PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                 PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion  PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                     PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                    PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                  PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                 PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                  PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                 PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                      00000000011111111112222222222333
                      12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
      is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
```

```
        is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
        is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern';
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);

// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
        (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
```

```
                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
} else {
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
  below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                    }
                }
            } else Gulliver = cbPattern-(2-1);
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
}
*/
```

```
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00    mov r8d, 65536
//  00452 44 89 5c 24 20    mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60    mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff    call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
                //"stes"
                //"test"
                //"est "
                //"st f"
                //"t fo"
                //" fox"
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
                //for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                // Above line is replaced by next one with better hashing:
                for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1;
                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
                        // Above line is replaced by next one with better hashing:
                        if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
```

```
       #1
                                                      //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
       0 ) Gulliver = cbPattern-(2-1)-2-4; else {
                                                      // Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
       the middle DWORD is 'bc'.
                                                      // The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
       + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                      // Above line is replaced by next one with better hashing:
                                                      // When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
       for ASCII texts with no symbols in range 128-255.
                                                      // Example for genomesque pair TT+TT being shifted by (16-1):
                                                      // T           = 01010100
                                                      // TT          = 01010100 01010100
                                                      // TTTT        = 01010100 01010100 01010100 01010100
                                                      // TTTT>>16    = 00000000 00000000 01010100 01010100
                                                      // TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
       0 for English texts & 'ACGT' data.
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
                                                      // 'Maximus' uses branched 'if', again.
                                                      if ( \
                                                      ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
       1 )] ) == 0 \
                                                      || ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
       1 )] ) == 0 \
                                                      ) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
                                                      // Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                         |
//     DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!    |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = -------------------------  |
//         DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!       |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ----------------------------  |
//             DWORD #1                                                                     |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                  |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = -------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                         |
//     DWORD #3                                                                        |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!    |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = -------------------------  |
//         DWORD #2                                                                       |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!       |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ----------------------------  |
//             DWORD #1                                                                     |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                  |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = -------------------------------
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
       1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
       *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
                                                      //if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
       + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
                                                                   if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                                                // Order 4 [
                                                                   // Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
                                                                   // Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
       'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                                                                   //0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
                                                                   //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                                   //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                                   //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                                   //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                                   //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                                   //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                                   //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                                                count = cbPattern-4+1;
                                                                                // Below comparison is UNIdirectional:
                                                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                                             count = count-4;

       if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
```

```
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }


                                                // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                              for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                      if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                      if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
    = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
    *)&pbTarget[i] == ulHashPattern)
//                                              }
//                                              if ( count <= 0 ) return(pbTarget+i);
                                                        // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it
    should
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,
    as it should
                                                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or
    bigger, as it should
                                                //      if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short
    *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                        // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                        // Needle:    fastest tax
                                                        // Window: ...fastast tax...
                                                        // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                        // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.
    Gulliver remains 1.
                                                        // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even
    not in ACGT data, that's why I commented it in original 'Shockeroo'.
                                                        //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )
    & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                        // Above line is replaced by next one with better hashing:
//                                                      if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1]>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
    1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                // Order 4 ]
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
    } //if ( cbPattern<4 )
}

char * Railgun_Trolldom_f (char * pbTarget, char * pbPattern, uint32_t cbTarget, uint32_t cbPattern)
{
    char * pbTargetMax = pbTarget + cbTarget;
    uint32_t ulHashPattern;
    signed long count;

    unsigned char bm_Horspool_Order2[256*256]; // Bitwise soon...
    unsigned char bm_Horspool_Order2bitwise[(256*256)>>3]; // Bitwise soon...
    uint32_t i, Gulliver;
```

```
      uint32_t PRIMALposition, PRIMALpositionCANDIDATE;
      uint32_t PRIMALlength, PRIMALlengthCANDIDATE;
      uint32_t j, FoundAtPosition;

// Quadruplet [
    //char * pbTargetMax = pbTarget + cbTarget;
    //register unsigned long  ulHashPattern;
    uint32_t ulHashTarget;
    //unsigned long count;
    uint32_t countSTATIC;
    unsigned char SINGLET;
    uint32_t Quadruplet2nd;
    uint32_t Quadruplet3rd;
    uint32_t Quadruplet4th;
    uint32_t AdvanceHopperGrass;
// Quadruplet ]

      if (cbPattern > cbTarget) return(NULL);

      if ( cbPattern<4 ) {
            // SSE2 i.e. 128bit Assembly rules here, Mischa knows best:
            // ...
            pbTarget = pbTarget+cbPattern;
            ulHashPattern = ( (*(char *)(pbPattern))<<8 ) + *(pbPattern+(cbPattern-1));
            if ( cbPattern==3 ) {
                        for ( ;; ) {
                                    if ( ulHashPattern == ( (*(char *)(pbTarget-3))<<8 ) + *(pbTarget-1) ) {
                                                if ( *(char *)(pbPattern+1) == *(char *)(pbTarget-2) ) return((pbTarget-3));
                                    }
                                    if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) {
                                                pbTarget++;
                                                if ( (char)(ulHashPattern>>8) != *(pbTarget-2) ) pbTarget++;
                                    }
                                    pbTarget++;
                                    if (pbTarget > pbTargetMax) return(NULL);
                        }
            } else {
            }
            for ( ;; ) {
                        if ( ulHashPattern == ( (*(char *)(pbTarget-2))<<8 ) + *(pbTarget-1) ) return((pbTarget-2));
                        if ( (char)(ulHashPattern>>8) != *(pbTarget-1) ) pbTarget++;
                        pbTarget++;
                        if (pbTarget > pbTargetMax) return(NULL);
            }
      } else { //if ( cbPattern<4 )
            if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

// This is the awesome 'Railgun_Quadruplet', it did outperform EVERYWHERE the fastest strstr (back in old GLIBCes ~2003, by the Dutch hacker Stephen R. van den Berg), suitable for short haystacks ~100bytes.
// Caution: For better speed the case 'if (cbPattern==1)' was removed, so Pattern must be longer than 1 char.
// char * Railgun_Quadruplet (char * pbTarget, char * pbPattern, unsigned long cbTarget, unsigned long cbPattern)
// ...
//    if (cbPattern > cbTarget) return(NULL);
//} else { //if ( cbPattern<4)
if (cbTarget<777) // This value is arbitrary(don't know how exactly), it ensures(at least must) better performance than 'Boyer_Moore_Horspool'.
{
       pbTarget = pbTarget+cbPattern;
       ulHashPattern = *(uint32_t *)(pbPattern);
//        countSTATIC = cbPattern-1;

    //SINGLET = *(char *)(pbPattern);
    SINGLET = ulHashPattern & 0xFF;
    Quadruplet2nd = SINGLET<<8;
    Quadruplet3rd = SINGLET<<16;
    Quadruplet4th = SINGLET<<24;

    for ( ;; )
    {
      AdvanceHopperGrass = 0;
      ulHashTarget = *(uint32_t *)(pbTarget-cbPattern);

       if ( ulHashPattern == ulHashTarget ) { // Three unnecessary comparisons here, but 'AdvanceHopperGrass' must be calculated - it has a higher priority.
//         count = countSTATIC;
//         while ( count && *(char *)(pbPattern+1+(countSTATIC-count)) == *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) {
//            if ( countSTATIC==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-cbPattern+1+(countSTATIC-count)) ) AdvanceHopperGrass++;
//               count--;
//         }
        count = cbPattern-1;
        while ( count && *(char *)(pbPattern+(cbPattern-count)) == *(char *)(pbTarget-count) ) {
             if ( cbPattern-1==AdvanceHopperGrass+count && SINGLET != *(char *)(pbTarget-count) ) AdvanceHopperGrass++;
              count--;
        }
        if ( count == 0) return((pbTarget-cbPattern));
       } else { // The goal here: to avoid memory accesses by stressing the registers.
    if ( Quadruplet2nd != (ulHashTarget & 0x0000FF00) ) {
        AdvanceHopperGrass++;
        if ( Quadruplet3rd != (ulHashTarget & 0x00FF0000) ) {
             AdvanceHopperGrass++;
             if ( Quadruplet4th != (ulHashTarget & 0xFF000000) ) AdvanceHopperGrass++;
```

```
     }
  }
    }

    AdvanceHopperGrass++;

    pbTarget = pbTarget + AdvanceHopperGrass;
     if (pbTarget > pbTargetMax)
        return(NULL);
  }
} else if (cbTarget<77777) { // The warmup/overhead is lowered from 64K down to 8K, however the bitwise additional instructions quickly start hurting the throughput/traversal.
// The below bitwise 0|1 BMH2 gives 1427 bytes/s for 'Don_Quixote' with Intel:
// The below bitwise 0|1 BMH2 gives 1242 bytes/s for 'Don_Quixote' with GCC:
//    } else { //if ( cbPattern<4 )
//      if ( cbPattern<=NeedleThreshold2vs4Decumanus ) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                //for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < (256*256)>>3; i++) {bm_Horspool_Order2bitwise[i]=0;}
                //for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                for (i=0; i < cbPattern-2+1; i++) bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3]= bm_Horspool_Order2bitwise[(*(unsigned short *)(pbPattern+i))>>3] |
    (1<<((*(unsigned short *)(pbPattern+i))&0x7));
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                        if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1])&0x7)) ) != 0 ) {
                                //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                if ( ( bm_Horspool_Order2bitwise[(*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])>>3] & (1<<((*(unsigned short *)&pbTarget[i+cbPattern-1-1-2])&0x7)) )
    == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
//      } else { // if ( cbPattern<=NeedleThreshold2vs4Decumanus )
} else { //if (cbTarget<777)
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

// Slower than Swampshine's simple 0|1 segment:
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
  FoundAtPosition = cbPattern;
  PRIMALpositionCANDIDATE=i;
  while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
     j = PRIMALpositionCANDIDATE + 1;
     while ( j <= (FoundAtPosition-1) ) {
       if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
       j++;
     }
     PRIMALpositionCANDIDATE++;
  }
  PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
  if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
if (cbPattern<4) {
```

```
    cbPattern = PRIMALlengthCANDIDATE;
    pbPattern = pbPattern - (PRIMALposition-1);
}
if (cbPattern == PRIMALlengthCANDIDATE) {
                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;
                                                if ( count <= 0 ) return(pbTarget+i);
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
} else { //if (cbPattern == PRIMALlengthCANDIDATE) {
// BMH Order 2 [
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                // The above 'for' gives  1424 bytes/s for 'Don_Quixote' with Intel:
                // The above 'for' gives  1431 bytes/s for 'Don_Quixote' with GCC:
                // The below 'memset' gives  1389 bytes/s for 'Don_Quixote' with Intel:
                // The below 'memset' gives  1432 bytes/s for 'Don_Quixote' with GCC:
                //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                        if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back
    (Order-1) chars i.e. one char for Order 2
                        if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                        count = cbPattern-4+1;
                                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                }
                                Gulliver = 1;
                        } else
                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                        }
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
// BMH Order 2 ]
} //if (cbPattern == PRIMALlengthCANDIDATE) {
*/

/*
So the result on Core 2 Q9550s @2.83GHz:
```

```
¦ testfile\Searcher                      ¦ GNU/GLIBC memmem()    ¦ Railgun_Swampshine    ¦ Railgun_Trolldom      ¦
¦--------------------------------------------------------------------------------------¦-----------------------¦
¦ Compiler                               ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦ Intel 15.0 ¦ GCC 5.10 ¦
¦--------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Don     ¦        190 ¦      226 ¦       1654 ¦     1729 ¦       1147 ¦     1764 ¦
¦ _Quixote_996_(ANSI).txt                ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 2,347,772 bytes                        ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦--------------------------------------------------------------------------------------¦-----------------------¦
¦ The_Project_Gutenberg_EBook_of_Dokoe   ¦        582 ¦      760 ¦       3094 ¦     2898 ¦       2410 ¦     3036 ¦
¦ _by_Hakucho_Masamune_(Japanese_UTF-8).txt ¦         ¦          ¦            ¦          ¦            ¦          ¦
¦ 899,425 bytes                          ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦--------------------------------------------------------------------------------------¦-----------------------¦
¦ Dragonfly_genome_shotgun_sequence      ¦        104 ¦      109 ¦        445 ¦      458 ¦        484 ¦      553 ¦
¦ _(ACGT_alphabet).fasta                 ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦ 4,487,433 bytes                        ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦--------------------------------------------------------------------------------------¦-----------------------¦
¦ LAOTZU_Wu_Wei_(BINARY).pdf             ¦         99 ¦      144 ¦        629 ¦      580 ¦        185 ¦      570 ¦
¦ 954,035 bytes                          ¦            ¦          ¦            ¦          ¦            ¦          ¦
¦--------------------------------------------------------------------------------------¦-----------------------¦
Below segment (when compiled with Intel) is very slow, see Railgun_Trolldom two sub-columns above, compared to GCC:
*/
/*
// BMH Order 2 [
                        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
                        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= (cbPattern-1);} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
                        // The above 'for' is translated by Intel as:
//.B5.21::
//  0013f 83 c0 40          add eax, 64
//  00142 66 0f 7f 44 14
//        60                movdqa XMMWORD PTR [96+rsp+rdx], xmm0
//  00148 3d 00 00 01 00    cmp eax, 65536
//  0014d 66 0f 7f 44 14
//        70                movdqa XMMWORD PTR [112+rsp+rdx], xmm0
//  00153 66 0f 7f 84 14
//        80 00 00 00       movdqa XMMWORD PTR [128+rsp+rdx], xmm0
//  0015c 66 0f 7f 84 14
//        90 00 00 00       movdqa XMMWORD PTR [144+rsp+rdx], xmm0
//  00165 89 c2             mov edx, eax
//  00167 72 d6             jb .B5.21
                        //memset(&bm_Horspool_Order2[0], cbPattern-1, 256*256); // Why why? It is 1700:1000 slower than above 'for'!?
                        // The above 'memset' is translated by Intel as:
//  00127 41 b8 00 00 01
//        00                mov r8d, 65536
//  0012d 44 8b 26          mov r12d, DWORD PTR [rsi]
//  00130 e8 fc ff ff ff    call _intel_fast_memset
                        // ! The problem is that 256*256, 64KB, is already too much, going bitwise i.e. 8KB is not that better, when 'cbPattern-1' is bigger than 255 - an unsigned char - then
                        // we must switch to 0¦1 table i.e. present or not. Since we are in 'if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {' branch and NeedleThreshold2vs4swampLITE, by default, is 19 - it is okay to use 'memset'. !
                        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
                        i=0;
                        while (i <= cbTarget-cbPattern) {
                                    Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
                                    if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
                                    if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                                                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                                                            count = cbPattern-4+1;
                                                            while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                        count = count-4;
                                                            if ( count <= 0 ) return(pbTarget+i);
                                                }
                                                Gulliver = 1;
                                    } else
                                                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
                                    }
                                    i = i + Gulliver;
                                    //GlobalI++; // Comment it, it is only for stats.
                        }
                        return(NULL);
// BMH Order 2 ]
*/
// Above fragment in Assembly:
/*
; mark_description "Intel(R) C++ Intel(R) 64 Compiler XE for applications running on Intel(R) 64, Version 15.0.0.108 Build 20140";
; mark_description "-O3 -QxSSE2 -D_N_XMM -D_N_prefetch_4096 -D_N_Branchfull -D_N_HIGH_PRIORITY -FA";
       ALIGN     16
.B6.1::                         ; Preds .B6.0
        push      rbx                                           ;3435.1
        push      r13                                           ;3435.1
        push      r15                                           ;3435.1
        push      rbp                                           ;3435.1
        mov       eax, 65592                                    ;3435.1
        call      __chkstk                                      ;3435.1
        sub       rsp, 65592                                    ;3435.1
        cmp       r9d, r8d                                      ;3460.18
        ja        .B6.25        ; Prob 28%                      ;3460.18
                                ; LOE rdx rcx rbx rsi rdi r12 r14 r8d r9d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.3::                         ; Preds .B6.1
        mov       r13d, DWORD PTR [rdx]                         ;3491.33
```

```
        lea       ebp, DWORD PTR [-1+r9]                        ;3492.67
        movzx     eax, bpl                                      ;3492.67
        xor       r10d, r10d                                    ;3492.4
        movd      xmm0, eax                                     ;3492.67
        xor       eax, eax                                      ;3492.4
        punpcklbw xmm0, xmm0                                    ;3492.67
        punpcklwd xmm0, xmm0                                    ;3492.67
        punpckldq xmm0, xmm0                                    ;3492.67
        punpcklqdq xmm0, xmm0                                   ;3492.67
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.4::                         ; Preds .B6.4 .B6.3
        add       eax, 64                                       ;3492.4
        movdqa    XMMWORD PTR [48+rsp+r10], xmm0                ;3492.33
        cmp       eax, 65536                                    ;3492.4
        movdqa    XMMWORD PTR [64+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [80+rsp+r10], xmm0                ;3492.33
        movdqa    XMMWORD PTR [96+rsp+r10], xmm0                ;3492.33
        mov       r10d, eax                                     ;3492.4
        jb        .B6.4         ; Prob 99%                      ;3492.4
                                ; LOE rdx rcx rbx rsi rdi r10 r12 r14 eax ebp r8d r9d r13d xmm0 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.5::                         ; Preds .B6.4
        test      ebp, ebp                                      ;3515.28
        je        .B6.12        ; Prob 50%                      ;3515.28
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.6::                         ; Preds .B6.5
        mov       eax, 1                                        ;3515.4
        lea       r11d, DWORD PTR [-1+r9]                       ;3515.4
        mov       r15d, r11d                                    ;3515.4
        xor       r10d, r10d                                    ;3515.4
        shr       r15d, 1                                       ;3515.4
        test      r15d, r15d                                    ;3515.4
        jbe       .B6.10        ; Prob 15%                      ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.8::                         ; Preds .B6.6 .B6.8
        lea       eax, DWORD PTR [r10+r10]                      ;3515.36
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.36
        inc       r10d                                          ;3515.4
        cmp       r10d, r15d                                    ;3515.4
        movzx     ebx, WORD PTR [rax+rdx]                       ;3515.75
        mov       BYTE PTR [48+rsp+rbx], al                     ;3515.36
        jb        .B6.8         ; Prob 64%                      ;3515.4
                                ; LOE rdx rcx rsi rdi r12 r14 ebp r8d r9d r10d r11d r13d r15d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.9::                         ; Preds .B6.8
        lea       eax, DWORD PTR [1+r10+r10]                    ;3515.4
                                ; LOE rdx rcx rbx rsi rdi r12 r14 eax ebp r8d r9d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.10::                        ; Preds .B6.9 .B6.6
        dec       eax                                           ;3515.36
        cmp       eax, r11d                                     ;3515.4
        jae       .B6.12        ; Prob 15%                      ;3515.4
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.11::                        ; Preds .B6.10
        movzx     r10d, WORD PTR [rax+rdx]                      ;3515.75
        mov       BYTE PTR [48+rsp+r10], al                     ;3515.36
                                ; LOE rdx rcx rbx rsi rdi r12 r14 ebp r8d r9d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.12::                        ; Preds .B6.5 .B6.10 .B6.11
        xor       r10d, r10d                                    ;3516.4
        lea       r15d, DWORD PTR [-3+r9]                       ;3522.27
        movsxd    r15, r15d                                     ;3522.7
        sub       r8d, r9d                                      ;3517.16
        lea       r11d, DWORD PTR [-2+r9]                       ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.13::                        ; Preds .B6.12 .B6.24
        lea       eax, DWORD PTR [-2+r9+r10]                    ;3518.78
        movzx     ebx, WORD PTR [rax+rcx]                       ;3518.55
        movzx     eax, BYTE PTR [48+rsp+rbx]                    ;3518.16
        cmp       eax, ebp                                      ;3519.32
        je        .B6.24        ; Prob 50%                      ;3519.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.14::                        ; Preds .B6.13
        cmp       eax, r11d                                     ;3520.32
        jne       .B6.23        ; Prob 62%                      ;3520.32
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.15::                        ; Preds .B6.14
        mov       eax, r10d                                     ;3521.25
        add       rax, rcx                                      ;3521.25
        cmp       r13d, DWORD PTR [rax]                         ;3521.40
        je        .B6.17        ; Prob 50%                      ;3521.40
                                ; LOE rax rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.16::                        ; Preds .B6.26 .B6.15
        mov       eax, 1                                        ;3527.6
        jmp       .B6.24        ; Prob 100%                     ;3527.6
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.17::                        ; Preds .B6.15
        mov       rbx, r15                                      ;3522.7
        test      r15, r15                                      ;3523.23
        jle       .B6.22        ; Prob 2%                       ;3523.23
                                ; LOE rax rdx rcx rbx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
```

```
.B6.18::                        ; Preds .B6.17
        mov       QWORD PTR [32+rsp], rsi                       ;
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.19::                        ; Preds .B6.20 .B6.18
        mov       esi, DWORD PTR [-1+rbx+rdx]                   ;3523.58
        cmp       esi, DWORD PTR [-1+rbx+rax]                   ;3523.79
        jne       .B6.26        ; Prob 20%                      ;3523.79
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.20::                        ; Preds .B6.19
        add       rbx, -4                                       ;3524.22
        test      rbx, rbx                                      ;3523.23
        jg        .B6.19        ; Prob 82%                      ;3523.23
                                ; LOE rax rdx rcx rbx rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.21::                        ; Preds .B6.20
        mov       rsi, QWORD PTR [32+rsp]                       ;
                                ; LOE rax rbx rsi rdi r12 r14 xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.22::                        ; Preds .B6.17 .B6.21
        add       rsp, 65592                                    ;3525.32
        pop       rbp                                           ;3525.32
        pop       r15                                           ;3525.32
        pop       r13                                           ;3525.32
        pop       rbx                                           ;3525.32
        ret                                                     ;3525.32
                                ; LOE
.B6.23::                        ; Preds .B6.14
        neg       eax                                           ;3529.17
        add       eax, r9d                                      ;3529.17
        add       eax, -2                                       ;3529.40
                                ; LOE rdx rcx rsi rdi r12 r14 r15 eax ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.24::                        ; Preds .B6.16 .B6.23 .B6.13
        add       r10d, eax                                     ;3531.13
        cmp       r10d, r8d                                     ;3517.25
        jbe       .B6.13        ; Prob 82%                      ;3517.25
                                ; LOE rdx rcx rsi rdi r12 r14 r15 ebp r8d r9d r10d r11d r13d xmm6 xmm7 xmm8 xmm9 xmm10 xmm11 xmm12 xmm13 xmm14 xmm15
.B6.25::                        ; Preds .B6.1 .B6.24
        xor       eax, eax                                      ;3534.10
        add       rsp, 65592                                    ;3534.10
        pop       rbp                                           ;3534.10
        pop       r15                                           ;3534.10
        pop       r13                                           ;3534.10
        pop       rbx                                           ;3534.10
        ret                                                     ;3534.10
                                ; LOE
.B6.26::                        ; Preds .B6.19                  ; Infreq
        mov       rsi, QWORD PTR [32+rsp]                       ;
        jmp       .B6.16        ; Prob 100%                     ;
*/

// GCC 5.10; >gcc -O3 -m64 -fomit-frame-pointer
/*
Railgun_Trolldom:
        pushq %r15
        .seh_pushreg    %r15
        movl  $65592, %eax
        pushq %r14
        .seh_pushreg    %r14
        pushq %r13
        .seh_pushreg    %r13
        pushq %r12
        .seh_pushreg    %r12
        pushq %rbp
        .seh_pushreg    %rbp
        pushq %rdi
        .seh_pushreg    %rdi
        pushq %rsi
        .seh_pushreg    %rsi
        pushq %rbx
        .seh_pushreg    %rbx
        call  ___chkstk_ms
        subq  %rax, %rsp
        .seh_stackalloc 65592
        .seh_endprologue
        cmpl  %r9d, %r8d
        movq  %rcx, %rbx
        movq  %rdx, %rdi
        movl  %r8d, %r12d
        movl  %r9d, %esi
        jb    .L118
        movl  (%rdx), %ebp
        leal  -1(%r9), %edx
        movl  $65536, %r8d
        leaq  48(%rsp), %rcx
        movzbl%dl, %edx
        call  memset
        movl  %esi, %r11d
        subl  $1, %r11d
        je    .L119
        xorl  %eax, %eax
        .p2align 4,,10
```

```
.L113:
	movzwl(%rdi,%rax), %edx
	movb	%al, 48(%rsp,%rdx)
	addq	$1, %rax
	cmpl	%eax, %r11d
	ja	.L113
.L112:
	leal	-4(%rsi), %r9d
	movl	%r12d, %r8d
	xorl	%edx, %edx
	leal	-3(%rsi), %eax
	shrl	$2, %r9d
	subl	%esi, %r8d
	leal	-2(%rsi), %r10d
	movslq%eax, %r14
	negq	%r9
	movl	%eax, 44(%rsp)
	leaq	-1(%r14), %r15
	salq	$2, %r9
	leaq	(%rdi,%r14), %r13
	jmp	.L117
	.p2align 4,,10
.L130:
	movl	%r10d, %eax
	subl	%ecx, %eax
	cmpl	%r10d, %ecx
	je	.L129
.L114:
	addl	%eax, %edx
	cmpl	%r8d, %edx
	ja	.L118
.L117:
	leal	(%rdx,%r10), %eax
	movzwl(%rbx,%rax), %eax
	movzbl48(%rsp,%rax), %ecx
	cmpl	%r11d, %ecx
	jne	.L130
	movl	%r11d, %eax
	addl	%eax, %edx
	cmpl	%r8d, %edx
	jbe	.L117
.L118:
	xorl	%eax, %eax
	jmp	.L128
	.p2align 4,,10
.L129:
	movl	%edx, %ecx
	movl	$1, %eax
	leaq	(%rbx,%rcx), %r12
	cmpl	(%r12), %ebp
	jne	.L114
	movl	44(%rsp), %esi
	testl	%esi, %esi
	jle	.L124
	movl	(%r12,%r15), %esi
	cmpl	%esi, (%rdi,%r15)
	jne	.L114
	addq	%r14, %rcx
	xorl	%eax, %eax
	addq	%rbx, %rcx
	jmp	.L116
	.p2align 4,,10
.L132:
	movl	-5(%r13,%rax), %esi
	subq	$4, %rax
	cmpl	-1(%rcx,%rax), %esi
	jne	.L131
.L116:
	cmpq	%rax, %r9
	jne	.L132
.L124:
	movq	%r12, %rax
.L128:
	addq	$65592, %rsp
	popq	%rbx
	popq	%rsi
	popq	%rdi
	popq	%rbp
	popq	%r12
	popq	%r13
	popq	%r14
	popq	%r15
	ret
	.p2align 4,,10
.L131:
	movl	$1, %eax
	jmp	.L114
.L119:
	xorl	%r11d, %r11d
```

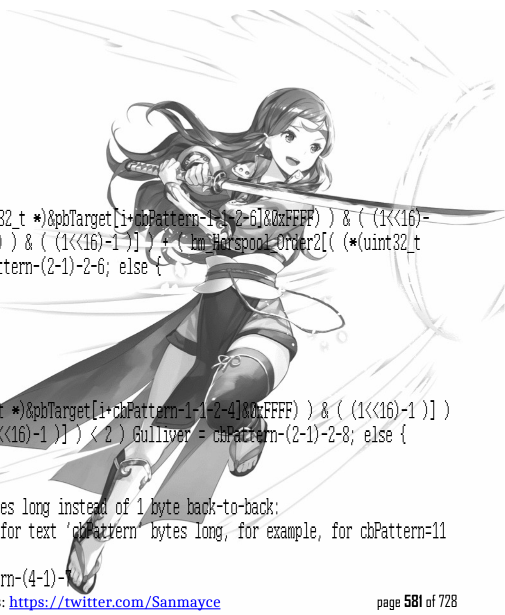
```
       jmp   .L112
*/

} //if (cbTarget<777)

             } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=29 PRIMALlength=04  'vvvv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=08 PRIMALlength=20  'vvvBOOMSHAKALAKAvvvv'
// Needle: Trollland                                    PRIMALposition=01 PRIMALlength=09  'Trollland'
// Needle: Swampwalker                                  PRIMALposition=01 PRIMALlength=11  'Swampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=15  'licenselessness'
// Needle: alfalfa                                      PRIMALposition=02 PRIMALlength=06  'lfalfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=08  'Sandokan'
// Needle: shazamish                                    PRIMALposition=01 PRIMALlength=09  'shazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=06 PRIMALlength=20  'icius Simplicissimus'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=32  'domilliaquadringenquattuorquinqu'
// Needle: boom-boom                                    PRIMALposition=02 PRIMALlength=08  'oom-boom'
// Needle: vvvvv                                        PRIMALposition=01 PRIMALlength=04  'vvvv'
// Needle: 12345                                        PRIMALposition=01 PRIMALlength=05  '12345'
// Needle: likey-likey                                  PRIMALposition=03 PRIMALlength=09  'key-likey'
// Needle: BOOOOOM                                      PRIMALposition=03 PRIMALlength=05  'OOOOM'
// Needle: aaaaaBOOOOOM                                 PRIMALposition=02 PRIMALlength=09  'aaaaBOOOO'
// Needle: BOOOOOMaaaaa                                 PRIMALposition=03 PRIMALlength=09  'OOOOMaaaa'
PRIMALlength=0;
for (i=0+(1); i < cbPattern-((4)-1)+(1)-(1); i++) { // -(1) because the last BB (Building-Block) order 4 has no counterpart(s)
        FoundAtPosition = cbPattern - ((4)-1) + 1;
        PRIMALpositionCANDIDATE=i;
        while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
              j = PRIMALpositionCANDIDATE + 1;
              while ( j <= (FoundAtPosition-1) ) {
                           if ( *(uint32_t *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(uint32_t *)(pbPattern+j-(1)) ) FoundAtPosition = j;
                           j++;
              }
              PRIMALpositionCANDIDATE++;
        }
        PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+1 +((4)-1);
        if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
        if (cbPattern-i+1 <= PRIMALlength) break;
        if (PRIMALlength > 128) break; // Bail Out for 129[+]
}
// Swampwalker_BAILOUT heuristic order 4 (Needle should be bigger than 4) ]

// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) [
// Needle: 1234567890qwertyuiopasdfghjklzxcv            PRIMALposition=01 PRIMALlength=33  '1234567890qwertyuiopasdfghjklzxcv'
// Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv            PRIMALposition=31 PRIMALlength=02  'vv'
// Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvvv            PRIMALposition=09 PRIMALlength=13  'vvBOOMSHAKALA'
// Needle: Trollland                                    PRIMALposition=05 PRIMALlength=05  'lland'
// Needle: Swampwalker                                  PRIMALposition=03 PRIMALlength=09  'ampwalker'
// Needle: licenselessness                              PRIMALposition=01 PRIMALlength=13  'licenselessne'
// Needle: alfalfa                                      PRIMALposition=04 PRIMALlength=04  'alfa'
// Needle: Sandokan                                     PRIMALposition=01 PRIMALlength=07  'Sandoka'
// Needle: shazamish                                    PRIMALposition=02 PRIMALlength=08  'hazamish'
// Needle: Simplicius Simplicissimus                    PRIMALposition=08 PRIMALlength=15  'ius Simplicissi'
// Needle: domilliaquadringenquattuorquinquagintillion PRIMALposition=01 PRIMALlength=19  'domilliaquadringenq'
// Needle: DODO                                         PRIMALposition=02 PRIMALlength=03  'ODO'
// Needle: DODOD                                        PRIMALposition=03 PRIMALlength=03  'DOD'
// Needle: aaaDODO                                      PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: aaaDODOD                                     PRIMALposition=02 PRIMALlength=05  'aaDOD'
// Needle: DODOaaa                                      PRIMALposition=02 PRIMALlength=05  'ODOaa'
// Needle: DODODaaa                                     PRIMALposition=03 PRIMALlength=05  'DODaa'
/*
PRIMALlength=0;
for (i=0+(1); i < cbPattern-2+1+(1)-(1); i++) { // -(1) because the last BB order 2 has no counterpart(s)
    FoundAtPosition = cbPattern;
    PRIMALpositionCANDIDATE=i;
    while ( PRIMALpositionCANDIDATE <= (FoundAtPosition-1) ) {
        j = PRIMALpositionCANDIDATE + 1;
        while ( j <= (FoundAtPosition-1) ) {
            if ( *(unsigned short *)(pbPattern+PRIMALpositionCANDIDATE-(1)) == *(unsigned short *)(pbPattern+j-(1)) ) FoundAtPosition = j;
            j++;
        }
        PRIMALpositionCANDIDATE++;
    }
    PRIMALlengthCANDIDATE = (FoundAtPosition-1)-i+(2);
    if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=i; PRIMALlength = PRIMALlengthCANDIDATE;}
}
*/
// Swampwalker_BAILOUT heuristic order 2 (Needle should be bigger than 2) ]

/*
Legend:
'[]' points to BB forming left or right boundary;
'{}' points to BB being searched for;
'()' position of duplicate and new right boundary;

                       00000000011111111112222222222333
```

```
                    12345678901234567890123456789012
Example #1 for Needle: 1234567890qwertyuiopasdfghjklzxcv  NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'
Example #2 for Needle: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  NewNeedle = 'vv'
Example #3 for Needle: vvvvvvvvvvBOOMSHAKALAKAvvvvvvvvv  NewNeedle = 'vvBOOMSHAKALA'


     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[12]34567890qwertyuiopasdfghjklzxc[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{12}]34567890qwertyuiopasdfghjklzxc[v?] ! Searching for '12', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_02: [1{2]3}4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
...
Step 01_30: [12]34567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_31: [12]34567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}1[23]4567890qwertyuiopasdfghjklzxc[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: 1[{23}]4567890qwertyuiopasdfghjklzxc[v?] ! Searching for '23', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_02: 1[2{3]4}567890qwertyuiopasdfghjklzxc[v?] ! Searching for '34', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
...
Step 02_29: 1[23]4567890qwertyuiopasdfghjkl{zx}c[v?] ! Searching for 'zx', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_30: 1[23]4567890qwertyuiopasdfghjklz{xc}[v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}1234567890qwertyuiopasdfghjklz[xc][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: 1234567890qwertyuiopasdfghjklz[{xc}][v?] ! Searching for 'xc', FoundAtPosition = 33, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=01 PRIMALlength=33, NewNeedle = '1234567890qwertyuiopasdfghjklzxcv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvvvvvvvvvvvvvvvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvvvvvvvvvvvvvvvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[vv][v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=31 PRIMALlength=02, NewNeedle = 'vv'



     PRIMALlength=00; FoundAtPosition=33;
Step 01_00: {}[vv]vvvvvvvvBOOMSHAKALAKAvvvvvvvv[v?] ! For position #01 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=01, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-01+(2)=33 !
Step 01_01: [{v(v}]v)vvvvvvvBOOMSHAKALAKAvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 02, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(02-1)-01+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
Step 02_00: {}v[vv]vvvvvvvBOOMSHAKALAKAvvvvvvvv[v?] ! For position #02 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=02, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-02+(2)=32 !
Step 02_01: v[{v(v}]v)vvvvvvBOOMSHAKALAKAvvvvvvvvv  ! Searching for 'vv', FoundAtPosition = 03, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(03-1)-02+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 09_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvv[v?] ! For position #09 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=09, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-09+(2)=25 !
Step 09_01: vvvvvvvv[{vv}]BOOMSHAKALAKA(vv)vvvvvvv  ! Searching for 'vv', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_02: vvvvvvvv[v{v]B}OOMSHAKALAKA[vv]vvvvvvv  ! Searching for 'vB', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_03: vvvvvvvv[vv]{BO}OMSHAKALAKA[vv]vvvvvvv  ! Searching for 'BO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_04: vvvvvvvv[vv]B{OO}MSHAKALAKA[vv]vvvvvvv  ! Searching for 'OO', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_05: vvvvvvvv[vv]BO{OM}SHAKALAKA[vv]vvvvvvv  ! Searching for 'OM', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_06: vvvvvvvv[vv]BOO{MS}HAKALAKA[vv]vvvvvvv  ! Searching for 'MS', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_07: vvvvvvvv[vv]BOOM{SH}AKALAKA[vv]vvvvvvv  ! Searching for 'SH', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_08: vvvvvvvv[vv]BOOMS{HA}KALAKA[vv]vvvvvvv  ! Searching for 'HA', FoundAtPosition = 24, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(24-1)-09+(2)=16 !
Step 09_09: vvvvvvvv[vv]BOOMSH{AK}AL(AK)Avvvvvvvvv  ! Searching for 'AK', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_10: vvvvvvvv[vv]BOOMSHA{KA}L[AK]Avvvvvvvvv  ! Searching for 'KA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_11: vvvvvvvv[vv]BOOMSHAK{AL}[AK]Avvvvvvvvv  ! Searching for 'AL', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
Step 09_12: vvvvvvvv[vv]BOOMSHAKA{L[A}K]Avvvvvvvvv  ! Searching for 'LA', FoundAtPosition = 21, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(21-1)-09+(2)=13 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
...
Step 31_00: {}vvvvvvvv[vv]BOOMSHAKALAKAvvvvvvvv[v?] ! For position #31 the initial boundaries are PRIMALpositionCANDIDATE=LeftBoundary=31, RightBoundary=FoundAtPosition-1, the CANDIDATE PRIMAL string length
       is RightBoundary-LeftBoundary+(2)=(33-1)-31+(2)=03 !
Step 31_01: vvvvvvvvvvBOOMSHAKALAKAvvvvvvv[{v(v}]v)  ! Searching for 'vv', FoundAtPosition = 32, PRIMALlengthCANDIDATE=RightBoundary-LeftBoundary+(2)=(32-1)-31+(2)=02 !
     if (PRIMALlengthCANDIDATE >= PRIMALlength) {PRIMALposition=PRIMALpositionCANDIDATE; PRIMALlength = PRIMALlengthCANDIDATE;}
     Result:
     PRIMALposition=09 PRIMALlength=13, NewNeedle = 'vvBOOMSHAKALA'
*/

// Here we have 4 or bigger NewNeedle, apply order 2 for pbPattern[i+(PRIMALposition-1)] with length 'PRIMALlength' and compare the pbPattern[i] with length 'cbPattern':
PRIMALlengthCANDIDATE = cbPattern;
cbPattern = PRIMALlength;
pbPattern = pbPattern + (PRIMALposition-1);
```

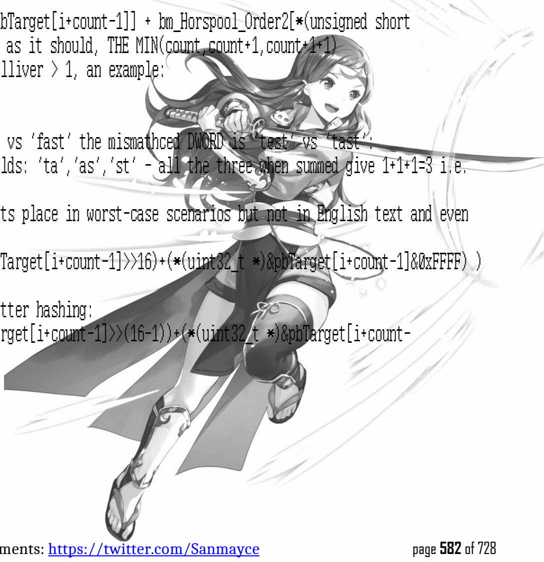
```
// Revision 2 commented section [
/*
if (cbPattern-1 <= 255) {
// BMH Order 2 [
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]= cbPattern-1;} // cbPattern-(Order-1) for Horspool; 'memset' if not optimized
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=i; // Rightmost appearance/position is needed
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]];
            if ( Gulliver != cbPattern-1 ) { // CASE #2: if equal means the pair (char order 2) is not found i.e. Gulliver remains intact, skip the whole pattern and fall back (Order-1) chars i.e. one char for Order 2
            if ( Gulliver == cbPattern-2 ) { // CASE #1: means the pair (char order 2) is found
                if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
                    count = cbPattern-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                        count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
        count = PRIMALlengthCANDIDATE-4+1;
        while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
            count = count-4;
        if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
    }
  }
}
                }
                Gulliver = 1;
            } else
                Gulliver = cbPattern - Gulliver - 2; // CASE #3: the pair is found and not as suffix i.e. rightmost position
            }
            i = i + Gulliver;
            //GlobalI++; // Comment it, it is only for stats.
        }
        return(NULL);
// BMH Order 2 ]
} else {
        // BMH order 2, needle should be >=4:
        ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
        for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
        for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
        i=0;
        while (i <= cbTarget-cbPattern) {
            Gulliver = 1; // 'Gulliver' is the skip
            if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                    if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
                        count = cbPattern-4+1;
                        while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                            count = count-4;
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
  // The line below is BUGGY:
  //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
  // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
  //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
  if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
    if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop below:
```

```
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);
}
*/
// Revision 2 commented section ]

        if ( cbPattern<=NeedleThreshold2vs4swampLITE ) {

                // BMH order 2, needle should be >=4:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // Above line is translated by Intel as:
//  0044c 41 b8 00 00 01
//        00            mov r8d, 65536
//  00452 44 89 5c 24 20  mov DWORD PTR [32+rsp], r11d
//  00457 44 89 54 24 60  mov DWORD PTR [96+rsp], r10d
//  0045c e8 fc ff ff ff  call _intel_fast_memset
                for (i=0; i < cbPattern-1; i++) bm_Horspool_Order2[*(unsigned short *)(pbPattern+i)]=1;
                i=0;
                while (i <= cbTarget-cbPattern) {
                        Gulliver = 1; // 'Gulliver' is the skip
                        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1]] != 0 ) {
                                if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+cbPattern-1-1-2]] == 0 ) Gulliver = cbPattern-(2-1)-2; else {
                                        if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                                                count = cbPattern-4+1;
                                                while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                        count = count-4;

    if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
    // The line below is BUGGY:
    //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
    // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
    //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
    if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
        if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
    below:
                count = PRIMALlengthCANDIDATE-4+1;
                while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                        count = count-4;
                if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
        }
    }
}
    } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                if ( count <= 0 ) return(pbTarget+i);
    }
                                        }
                                }
                        } else Gulliver = cbPattern-(2-1);
                        i = i + Gulliver;
                        //GlobalI++; // Comment it, it is only for stats.
                }
                return(NULL);

        } else { // if ( cbPattern<=NeedleThreshold2vs4swampLITE )

                // BMH pseudo-order 4, needle should be >=8+2:
                ulHashPattern = *(uint32_t *)(pbPattern); // First four bytes
//              for (i=0; i < 256*256; i++) {bm_Horspool_Order2[i]=0;}
                // In line below we "hash" 4bytes to 2bytes i.e. 16bit table, how to compute TOTAL number of BBs, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for
    example, for cbPattern=11 'fastest fox' and Order=4 we have BBs = 11-4+1=8:
                //"fast"
                //"aste"
```

```
			//"stes"
			//"test"
			//"est "
			//"st f"
			//"t fo"
			//" fox"
			//for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( *(unsigned short *)(pbPattern+i+0) + *(unsigned short *)(pbPattern+i+2) ) & ( (1<<16)-1 )]=1;
			//for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>16)+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
			// Above line is replaced by next one with better hashing:
			for (i=0; i < cbPattern-4+1; i++) bm_Horspool_Order2[( (*(uint32_t *)(pbPattern+i+0)>>(16-1))+(*(uint32_t *)(pbPattern+i+0)&0xFFFF) ) & ( (1<<16)-1 )]=1;
			i=0;
			while (i <= cbTarget-cbPattern) {
				Gulliver = 1;
				//if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD #1
				// Above line is replaced by next one with better hashing:
				if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2]&0xFFFF) ) & ( (1<<16)-1 )] != 0 ) { // DWORD
	#1
					//if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ==
	0 ) Gulliver = cbPattern-(2-1)-2-4; else {
					// Above line is replaced in order to strengthen the skip by checking the middle DWORD,if the two DWORDs are 'ab' and 'cd' i.e. [2x][2a][2b][2c][2d] then
	the middle DWORD is 'bc'.
					// The respective offsets (backwards) are: -10/-8/-6/-4 for 'xa'/'ab'/'bc'/'cd'.
					//if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-1 )] )
	+ ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
	*)&pbTarget[i+cbPattern-1-1-2-2]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
					// Above line is replaced by next one with better hashing:
					// When using (16-1) right shifting instead of 16 we will have two different pairs (if they are equal), the highest bit being lost do the job especialy
	for ASCII texts with no symbols in range 128-255.
					// Example for genomesque pair TT+TT being shifted by (16-1):
					// T           = 01010100
					// TT          = 01010100 01010100
					// TTTT        = 01010100 01010100 01010100 01010100
					// TTTT>>16    = 00000000 00000000 01010100 01010100
					// TTTT>>(16-1) = 00000000 00000000 10101000 10101000 <--- Due to the left shift by 1, the 8th bits of 1st and 2nd bytes are populated - usually they are
	0 for English texts & 'ACGT' data.
					//if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
	1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
	*)&pbTarget[i+cbPattern-1-1-2-2]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-4-2; else {
					// 'Maximus' uses branched 'if', again.
					if ( \
					( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6 +1]&0xFFFF) ) & ( (1<<16)-
	1 )] ) == 0 \
					|| ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]>>(16-1))+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4 +1]&0xFFFF) ) & ( (1<<16)-
	1 )] ) == 0 \
					) Gulliver = cbPattern-(2-1)-2-4-2 +1; else {
					// Above line is not optimized (several a SHR are used), we have 5 non-overlapping WORDs, or 3 overlapping WORDs, within 4 overlapping DWORDs so:
// [2x][2a][2b][2c][2d]
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]>>16) =     !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                       |
//     DWORD #3                                                                       |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16) =     !SHR to be avoided!   |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]>>16) =     !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                    |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
//
// So in order to remove 3 SHR instructions the equal extractions are:
// DWORD #4
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) =  !SHR to be avoided! <--
// [2x] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) =                      |
//     DWORD #3                                                                      |
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) =  !SHR to be avoided!  |<--
// [2a] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) = ------------------------  |
//         DWORD #2                                                                      |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) =  !SHR to be avoided!      |<--
// [2b] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) = ---------------------------  |
//             DWORD #1                                                                    |
// [2d] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]>>16) =                                |
// [2c] (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF) = ------------------------------
					//if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-6]&0xFFFF) ) & ( (1<<16)-
	1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] ) + ( bm_Horspool_Order2[( (*(uint32_t
	*)&pbTarget[i+cbPattern-1-1-2-0]&0xFFFF)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-2]&0xFFFF) ) & ( (1<<16)-1 )] ) < 3 ) Gulliver = cbPattern-(2-1)-2-6; else {
// Since the above Decumanus mumbo-jumbo (3 overlapping lookups vs 2 non-overlapping lookups) is not fast enough we go DuoDecumanus or 3x4:
// [2y][2x][2a][2b][2c][2d]
// DWORD #3
//         DWORD #2
//                 DWORD #1
					//if ( ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-4]&0xFFFF) ) & ( (1<<16)-1 )] )
	+ ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]>>16)+(*(uint32_t *)&pbTarget[i+cbPattern-1-1-2-8]&0xFFFF) ) & ( (1<<16)-1 )] ) < 2 ) Gulliver = cbPattern-(2-1)-2-8; else {
						if ( *(uint32_t *)&pbTarget[i] == ulHashPattern) {
							// Order 4 [
						// Let's try something "outrageous" like comparing with[out] overlap BBs 4bytes long instead of 1 byte back-to-back:
						// Inhere we are using order 4, 'cbPattern - Order + 1' is the number of BBs for text 'cbPattern' bytes long, for example, for cbPattern=11
	'fastest fox' and Order=4 we have BBs = 11-4+1=8:
						//0:"fast" if the comparison failed here, 'count' is 1; 'Gulliver' is cbPattern-(4-1)-7
```



```
                                                 //1:"aste" if the comparison failed here, 'count' is 2; 'Gulliver' is cbPattern-(4-1)-6
                                                 //2:"stes" if the comparison failed here, 'count' is 3; 'Gulliver' is cbPattern-(4-1)-5
                                                 //3:"test" if the comparison failed here, 'count' is 4; 'Gulliver' is cbPattern-(4-1)-4
                                                 //4:"est " if the comparison failed here, 'count' is 5; 'Gulliver' is cbPattern-(4-1)-3
                                                 //5:"st f" if the comparison failed here, 'count' is 6; 'Gulliver' is cbPattern-(4-1)-2
                                                 //6:"t fo" if the comparison failed here, 'count' is 7; 'Gulliver' is cbPattern-(4-1)-1
                                                 //7:" fox" if the comparison failed here, 'count' is 8; 'Gulliver' is cbPattern-(4-1)
                                                           count = cbPattern-4+1;
                                                           // Below comparison is UNIdirectional:
                                                           while ( count > 0 && *(uint32_t *)(pbPattern+count-1) == *(uint32_t *)(&pbTarget[i]+(count-1)) )
                                                                     count = count-4;

     if (cbPattern != PRIMALlengthCANDIDATE) { // No need of same comparison when Needle and NewNeedle are equal!
// count = cbPattern-4+1 = 23-4+1 = 20
// boomshakalakaZZZZZZ[ZZZZ] 20
// boomshakalakaZZ[ZZZZ]ZZZZ 20-4
// boomshakala[kaZZ]ZZZZZZZZ 20-8 = 12
// boomsha[kala]kaZZZZZZZZZZ 20-12 = 8
// boo[msha]kalakaZZZZZZZZZZ 20-16 = 4

// If we miss to hit then no need to compare the original: Needle
if ( count <= 0 ) {
// I have to add out-of-range checks...
// i-(PRIMALposition-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4

// "FIX" from 2014-Apr-27:
// Because (count-1) is negative, above fours are reduced to next twos:
// i-(PRIMALposition-1)+(count-1) >= 0
// &pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4
     // The line below is BUGGY:
     //if ( (i-(PRIMALposition-1) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) && (&pbTarget[i-(PRIMALposition-1)+(count-1)] <= pbTargetMax - 4) ) {
     // The line below is NOT OKAY, in fact so stupid, grrr, not a blunder, not carelessness, but overconfidence in writing "on the fly":
     //if ( ((signed int)(i-(PRIMALposition-1)+(count-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)] <= pbTargetMax - 4) ) {
// FIX from 2016-Aug-10 (two times failed to do simple boundary checks, pfu):
     if ( ((signed int)(i-(PRIMALposition-1)) >= 0) && (&pbTarget[i-(PRIMALposition-1)]+((PRIMALlengthCANDIDATE-4+1)-1) <= pbTargetMax - 4) ) {
          if ( *(uint32_t *)&pbTarget[i-(PRIMALposition-1)] == *(uint32_t *)(pbPattern-(PRIMALposition-1))) { // This fast check ensures not missing a match (for remainder) when going under 0 in loop
     below:
                    count = PRIMALlengthCANDIDATE-4+1;
                    while ( count > 0 && *(uint32_t *)(pbPattern-(PRIMALposition-1)+count-1) == *(uint32_t *)(&pbTarget[i-(PRIMALposition-1)]+(count-1)) )
                              count = count-4;
                    if ( count <= 0 ) return(pbTarget+i-(PRIMALposition-1));
          }
     }
}
     } else { //if (cbPattern != PRIMALlengthCANDIDATE)
                                                           if ( count <= 0 ) return(pbTarget+i);
     }

                                                           // In order to avoid only-left or only-right WCS the memcmp should be done as left-to-right and right-to-left AT THE SAME TIME.
                                                           // Below comparison is BIdirectional. It pays off when needle is 8+++ long:
//                                                         for (count = cbPattern-4+1; count > 0; count = count-4) {
//                                                                   if ( *(uint32_t *)(pbPattern+count-1) != *(uint32_t *)(&pbTarget[i]+(count-1)) ) {break;};
//                                                                   if ( *(uint32_t *)(pbPattern+(cbPattern-4+1)-count) != *(uint32_t *)(&pbTarget[i]+(cbPattern-4+1)-count) ) {count
     = (cbPattern-4+1)-count +(1); break;} // +(1) because two lookups are implemented as one, also no danger of 'count' being 0 because of the fast check outwith the 'while': if ( *(uint32_t
     *)&pbTarget[i] == ulHashPattern)
//                                                         }
//                                                         if ( count <= 0 ) return(pbTarget+i);
                                                                     // Checking the order 2 pairs in mismatched DWORD, all the 3:
                                                                     //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] == 0 ) Gulliver = count; // 1 or bigger, as it

     should

                                                                     //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1]] == 0 ) Gulliver = count+1; // 1 or bigger,

     as it should

                                                                     //if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] == 0 ) Gulliver = count+1+1; // 1 or

     bigger, as it should

                                                           //        if ( bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1]] + bm_Horspool_Order2[*(unsigned short

     *)&pbTarget[i+count-1+1]] + bm_Horspool_Order2[*(unsigned short *)&pbTarget[i+count-1+1+1]] < 3 ) Gulliver = count; // 1 or bigger, as it should, THE MIN(count,count+1,count+1+1)
                                                                     // Above compound 'if' guarantees not that Gulliver > 1, an example:
                                                                     // Needle:    fastest tax
                                                                     // Window: ...fastast tax...
                                                                     // After matching ' tax' vs ' tax' and 'fast' vs 'fast' the mismathced DWORD is 'test' vs 'tast':
                                                                     // 'tast' when factorized down to order 2 yields: 'ta','as','st' - all the three when summed give 1+1+1=3 i.e.

     Gulliver remains 1.

                                                                     // Roughly speaking, this attempt maybe has its place in worst-case scenarios but not in English text and even

     not in ACGT data, that's why I commented it in original 'Shockeroo'.

                                                                     //if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>16)+(*(uint32_t *)&pbTarget[i+count-1]&0xFFFF) )

     & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should

                                                                     // Above line is replaced by next one with better hashing:
//                                                                   if ( bm_Horspool_Order2[( (*(uint32_t *)&pbTarget[i+count-1])>>(16-1))+(*(uint32_t *)&pbTarget[i+count-
     1]&0xFFFF) ) & ( (1<<16)-1 )] == 0 ) Gulliver = count; // 1 or bigger, as it should
                                                           // Order 4 ]
                                                 }
                                       }
                             } else Gulliver = cbPattern-(2-1)-2; // -2 because we check the 4 rightmost bytes not 2.
                             i = i + Gulliver;
                             //GlobalI++; // Comment it, it is only for stats.
                   }
```



```
                                return(NULL);

                } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
                } // if ( cbPattern<=NeedleThreshold2vs4swampLITE )
        } //if ( cbPattern<4 )
}

void * memchrKAZE (const void * buf, const void * chr, unsigned long cnt)
{
        while ( cnt && (*(unsigned char *)buf != *(unsigned char*)chr) ) {
                buf = (unsigned char *)buf + 1;
                cnt--;
        }

        return(cnt ? (void *)buf : NULL);
}
//Exit:
//       returns pointer to first occurence of chr in buf
//       returns NULL if chr not found in the first cnt bytes


long KAZE_strlen (const char * str)
{
        const char *eos = str;

        while( *eos++ ) ;

        return( (int)(eos - str - 1) );
}
//_KAZE_strlen PROC NEAR
//; Line 225: const char *eos = str;
//        mov     ecx, DWORD PTR _str$[esp-4]
//        mov     eax, ecx
//$L725:
//; Line 227: while( *eos++ ) ;
//        mov     dl, BYTE PTR [eax]
//        inc     eax
//        test    dl, dl
//        jne     SHORT $L725
//; Line 229: return( (int)(eos - str - 1) );
//        sub     eax, ecx
//        dec     eax
//; Line 230
//        ret     0
//_KAZE_strlen ENDP

long KAZE_strlenLF (const char * str)
{
        const char *eos = str;
        char LFa[1];
        LFa[0] = 10; //BUG UNcrushed yet: for Windows 13 for POSIX 10
        while( *eos++ != LFa[0] ) ;

        return( (int)(eos - str - 1) );
}

//       wildcard ’*’ any character(s) or empty,
//       wildcard ’.’ any ALPHA character(s) or empty,
//       wildcard ’%’ any NON-ALPHA character(s) or empty,
//       wildcard ’@’/’#’ any character {or empty}/{and not empty},
//       wildcard ’^’/’$’ any ALPHA character {or empty}/{and not empty},
//       wildcard ’¦’/’~’ any NON-ALPHA character {or empty}/{and not empty},
//       TO-DO: wildcard ’+’/’`’ any WORD {or empty}/{and not empty}.

// Note: Due to different line endings(CRLF in Windows; LF in UNIX) you must add a ’¦’ wildcard in place of CR: for example in case of searching for ’*.pdf’ write ’*.pdf¦’.
// Pattern example: *%%take@%%$$¦


// Igor Pavlov’s variant modified by Kaze
// ’&’ stands for "standard" ’*’, ’+’ stands for "standard" ’?’
static boolean Wildcard_IP(const char *mask, int maskPos, const char *name, int namePos)
{
char maskChar;
int maskLen = maskGLOBALlen - maskPos;
int nameLen = nameGLOBALlen1 - namePos;
if (maskLen == 0) {
       if (nameLen == 0)
             return true;
       else
             return false;
}
maskChar = mask[maskPos];
if (maskChar == ’+’) { // and not empty
       /*
       if (Wildcard_IP(mask, maskPos + 1, name, namePos))
             return true;
       */
       if (nameLen == 0)
             return false;
```

```
        return Wildcard_IP(mask, maskPos + 1, name, namePos + 1);
} else if (maskChar == '&') {
        if (Wildcard_IP(mask, maskPos + 1, name, namePos))
              return true;
        if (nameLen == 0)
              return false;
        return Wildcard_IP(mask, maskPos, name, namePos + 1);
} else {
        if (maskChar != name[namePos])
              return false;
        return Wildcard_IP(mask, maskPos + 1, name, namePos + 1);
}
}
/*
; Compiler: Intel C++ Compiler XE for applications running on IA-32, Version 12.1
; Options: "-O3 -Qunroll";
_Wildcard_IP PROC NEAR PRIVATE
; parameter 1: eax
; parameter 2: edx
; parameter 3: ecx
; parameter 4: 72 + esp
.B6.1:

;;; {

  00000 56              push esi
  00001 57              push edi
  00002 53              push ebx
  00003 55              push ebp
  00004 83 ec 28        sub esp, 40
  00007 8b fa           mov edi, edx
  00009 89 44 24 10     mov DWORD PTR [16+esp], eax
  0000d f7 df           neg edi
  0000f 89 4c 24 14     mov DWORD PTR [20+esp], ecx
  00013 8b 44 24 48     mov eax, DWORD PTR [72+esp]

;;; char maskChar;
;;; int maskLen = maskGLOBALlen - maskPos;

  00017 8b e8           mov ebp, eax
  00019 89 7c 24 18     mov DWORD PTR [24+esp], edi
  0001d 8b f8           mov edi, eax
  0001f 8b 35 00 00 00
        00              mov esi, DWORD PTR [_maskGLOBALlen]
  00025 f7 df           neg edi

;;; int nameLen = nameGLOBALlen1 - namePos;

  00027 8b 0d 00 00 00
        00              mov ecx, DWORD PTR [_nameGLOBALlen1]
  0002d 89 74 24 1c     mov DWORD PTR [28+esp], esi
  00031 2b f2           sub esi, edx
  00033 89 4c 24 24     mov DWORD PTR [36+esp], ecx
  00037 89 54 24 20     mov DWORD PTR [32+esp], edx

.B6.2:
  0003b 8b 54 24 24     mov edx, DWORD PTR [36+esp]

;;; if (maskLen == 0)

  0003f 85 f6           test esi, esi
  00041 8d 1c 3a        lea ebx, DWORD PTR [edx+edi]
  00044 75 18           jne .B6.4 ; Prob 50%

.B6.3:
  00046 8b c3           mov eax, ebx
  00048 33 c9           xor ecx, ecx
  0004a ba 01 00 00 00  mov edx, 1
  0004f 85 c0           test eax, eax
  00051 0f 44 ca        cmove ecx, edx
  00054 8b c1           mov eax, ecx
  00056 83 c4 28        add esp, 40
  00059 5d              pop ebp
  0005a 5b              pop ebx
  0005b 5f              pop edi
  0005c 5e              pop esi
  0005d c3              ret

.B6.4:

;;;     if (nameLen == 0)
;;;           return true;
;;;     else
;;;           return false;
;;; maskChar = mask[maskPos];

  0005e 8b 4c 24 10     mov ecx, DWORD PTR [16+esp]
  00062 8b 54 24 20     mov edx, DWORD PTR [32+esp]
  00066 0f be 14 0a     movsx edx, BYTE PTR [edx+ecx]
```

```
;;; if (maskChar == '+') { // and not empty

  0006a 83 fa 2b        cmp edx, 43
  0006d 75 1c           jne .B6.7 ; Prob 67%

.B6.5:

;;;    /*
;;;    if (Wildcard_IP(mask, maskPos + 1, name, namePos))
;;;          return true;
;;;    */
;;;    if (nameLen == 0)

  0006f 85 db           test ebx, ebx
  00071 74 74           je .B6.14 ; Prob 5%

.B6.6:

;;;          return false;
;;;    return Wildcard_IP(mask, maskPos + 1, name, namePos + 1);

  00073 8b 4c 24 18     mov ecx, DWORD PTR [24+esp]
  00077 4f              dec edi
  00078 49              dec ecx
  00079 45              inc ebp
  0007a 8b 54 24 1c     mov edx, DWORD PTR [28+esp]
  0007e ff 44 24 20     inc DWORD PTR [32+esp]
  00082 89 4c 24 18     mov DWORD PTR [24+esp], ecx
  00086 8d 34 0a        lea esi, DWORD PTR [edx+ecx]
  00089 eb b0           jmp .B6.2 ; Prob 100%

.B6.7:
  0008b 83 fa 26        cmp edx, 38
  0008e 75 32           jne .B6.12 ; Prob 50%

.B6.8:

;;; } else if (maskChar == '&') {
;;;    if (Wildcard_IP(mask, maskPos + 1, name, namePos))

  00090 8b 54 24 20     mov edx, DWORD PTR [32+esp]
  00094 89 6c 24 0c     mov DWORD PTR [12+esp], ebp
  00098 8b c1           mov eax, ecx
  0009a 8b 4c 24 14     mov ecx, DWORD PTR [20+esp]
  0009e 8d 52 01        lea edx, DWORD PTR [1+edx]
  000a1 e8 fc ff ff ff  call _Wildcard_IP

.B6.19:
  000a6 0f b6 c0        movzx eax, al
  000a9 85 c0           test eax, eax
  000ab 75 08           jne .B6.11 ; Prob 28%

.B6.9:

;;;          return true;
;;;    if (nameLen == 0)

  000ad 85 db           test ebx, ebx
  000af 74 36           je .B6.14 ; Prob 5%

.B6.10:

;;;          return false;
;;;    return Wildcard_IP(mask, maskPos, name, namePos + 1);

  000b1 4f              dec edi
  000b2 45              inc ebp
  000b3 eb 86           jmp .B6.2 ; Prob 100%

.B6.11:
  000b5 b8 01 00 00 00  mov eax, 1
  000ba 83 c4 28        add esp, 40
  000bd 5d              pop ebp
  000be 5b              pop ebx
  000bf 5f              pop edi
  000c0 5e              pop esi
  000c1 c3              ret

.B6.12:

;;; } else {
;;;    if (maskChar != name[namePos])

  000c2 8b 4c 24 14     mov ecx, DWORD PTR [20+esp]
  000c6 3a 54 0d 00     cmp dl, BYTE PTR [ebp+ecx]
  000ca 75 1b           jne .B6.14 ; Prob 5%

.B6.13:
```

```
;;;         return false;
;;;    return Wildcard_IP(mask, maskPos + 1, name, namePos + 1);

  000cc 8b 4c 24 18      mov ecx, DWORD PTR [24+esp]
  000d0 4f               dec edi
  000d1 49               dec ecx
  000d2 45               inc ebp
  000d3 8b 54 24 1c      mov edx, DWORD PTR [28+esp]
  000d7 ff 44 24 20      inc DWORD PTR [32+esp]
  000db 89 4c 24 18      mov DWORD PTR [24+esp], ecx
  000df 8d 34 0a         lea esi, DWORD PTR [edx+ecx]
  000e2 e9 54 ff ff ff   jmp .B6.2 ; Prob 100%

.B6.14:
  000e7 33 c0            xor eax, eax
  000e9 83 c4 28         add esp, 40
  000ec 5d               pop ebp
  000ed 5b               pop ebx
  000ee 5f               pop edi
  000ef 5e               pop esi
  000f0 c3               ret
  000f1 90 8d b4 26 00
        00 00 00 8d bc
        27 00 00 00 00   ALIGN     16

_Wildcard_IP ENDP
*/


// Results on my laptop Core 2 T7500 2200MHz:
/*
D:\_KAZE\_KAZE_GOLD\Kazahana_source_EXEs_Benchmark\WildBench>_compile_Intel.bat

D:\_KAZE\_KAZE_GOLD\Kazahana_source_EXEs_Benchmark\WildBench>icl /O3 wildbench.c /FAcs /Fewildbench_Intel12
Intel(R) C++ Compiler XE for applications running on IA-32, Version 12.1.1.258 Build 20111011
Copyright (C) 1985-2011 Intel Corporation.  All rights reserved.

wildbench.c
Microsoft (R) Incremental Linker Version 10.00.30319.01
Copyright (C) Microsoft Corporation.  All rights reserved.

-out:wildbench_Intel12.exe
wildbench.obj

D:\_KAZE\_KAZE_GOLD\Kazahana_source_EXEs_Benchmark\WildBench>wildbench_Intel12.exe
Running three times, for charm ...

50000000 runs of WildcardMatch_Recursive_DezhiZhao  = 44.272s, r = 350000000
50000000 runs of WildcardMatch_Iterative_JackHandy  = 15.959s, r = 350000000
50000000 runs of WildcardMatch_Iterative_Kaze       = 18.237s, r = 350000000

50000000 runs of WildcardMatch_Recursive_DezhiZhao  = 46.035s, r = 350000000
50000000 runs of WildcardMatch_Iterative_JackHandy  = 14.711s, r = 350000000
50000000 runs of WildcardMatch_Iterative_Kaze       = 21.403s, r = 350000000

50000000 runs of WildcardMatch_Recursive_DezhiZhao  = 44.164s, r = 350000000
50000000 runs of WildcardMatch_Iterative_JackHandy  = 16.302s, r = 350000000
50000000 runs of WildcardMatch_Iterative_Kaze       = 18.595s, r = 350000000

D:\_KAZE\_KAZE_GOLD\Kazahana_source_EXEs_Benchmark\WildBench>_compile_VS2010.bat

D:\_KAZE\_KAZE_GOLD\Kazahana_source_EXEs_Benchmark\WildBench>cl /Ox wildbench.c /FAcs /Fewildbench_VS2010
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 16.00.30319.01 for 80x86
Copyright (C) Microsoft Corporation.  All rights reserved.

wildbench.c
Microsoft (R) Incremental Linker Version 10.00.30319.01
Copyright (C) Microsoft Corporation.  All rights reserved.

/out:wildbench_VS2010.exe
wildbench.obj

D:\_KAZE\_KAZE_GOLD\Kazahana_source_EXEs_Benchmark\WildBench>wildbench_VS2010.exe
Running three times, for charm ...

50000000 runs of WildcardMatch_Recursive_DezhiZhao  = 63.835s, r = 350000000
50000000 runs of WildcardMatch_Iterative_JackHandy  = 26.130s, r = 350000000
50000000 runs of WildcardMatch_Iterative_Kaze       = 26.567s, r = 350000000

50000000 runs of WildcardMatch_Recursive_DezhiZhao  = 62.384s, r = 350000000
50000000 runs of WildcardMatch_Iterative_JackHandy  = 26.005s, r = 350000000
50000000 runs of WildcardMatch_Iterative_Kaze       = 26.068s, r = 350000000

50000000 runs of WildcardMatch_Recursive_DezhiZhao  = 63.040s, r = 350000000
50000000 runs of WildcardMatch_Iterative_JackHandy  = 26.036s, r = 350000000
50000000 runs of WildcardMatch_Iterative_Kaze       = 26.333s, r = 350000000

D:\_KAZE\_KAZE_GOLD\Kazahana_source_EXEs_Benchmark\WildBench>
```

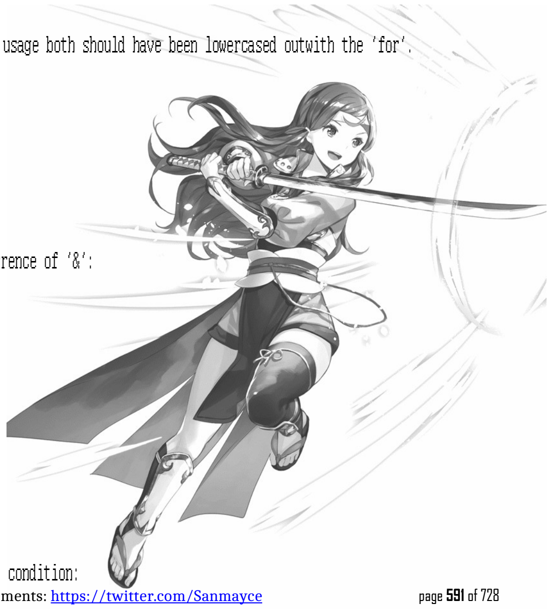
```
*/

// Igor Pavlov's recursive variant modified (and converted to iterative) by Kaze, 2013-Nov-28:
//static boolean Wildcard_IP(const char *mask, int maskPos, const char *name, int namePos)
//{
//int maskLen = maskGLOBALlen - maskPos;
//int nameLen = nameGLOBALlen - namePos;
//if (maskLen == 0)
//      if (nameLen == 0)
//            return true;
//      else
//            return false;
//if (mask[maskPos] == '*') {
//      if (Wildcard_IP(mask, maskPos + 1, name, namePos))
//            return true;
//      if (nameLen == 0)
//            return false;
//      return Wildcard_IP(mask, maskPos, name, namePos + 1);
//} else if (mask[maskPos] == '?') {
//      if (nameLen == 0)
//            return false;
//      return Wildcard_IP(mask, maskPos + 1, name, namePos + 1);
//} else {
//      if (mask[maskPos] != name[namePos])
//            return false;
//      return Wildcard_IP(mask, maskPos + 1, name, namePos + 1);
//}
//}
int WildcardMatch_Iterative_Kaze(const char* mask, const char* name) {
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
       if (*maskSTACK == '*') {
             mask = maskSTACK+1;
             if (!*mask) return 1;
             name = nameSTACK;
             BacktrackFlag = -1;
             goto Backtrack;
       }
       //else if (*maskSTACK == '?') {
       //} else {
       else if (*maskSTACK != '?') {
             //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
             if (*nameSTACK != *maskSTACK) {
                          if (!BacktrackFlag) return 0;
                          name++;
                          goto Backtrack;
             }
       }
}
while (*maskSTACK == '*') ++maskSTACK;
return (!*maskSTACK);
}
/*
; mark_description "Intel(R) C++ Compiler XE for applications running on IA-32, Version 12.1.1.258 Build 20111011";
; mark_description "-O3 -FAcs -Fewildbench";

_WildcardMatch_Iterative_Kaze          PROC NEAR
; parameter 1: 16 + esp
; parameter 2: 20 + esp
.B5.1:

;;; int WildcardMatch_Iterative_Kaze(const char* mask, const char* name) {

  00000 56             push esi
  00001 57             push edi
  00002 56             push esi
  00003 8b 4c 24 14    mov ecx, DWORD PTR [20+esp]
  00007 8b 44 24 10    mov eax, DWORD PTR [16+esp]

;;; const char* maskSTACK;
;;; const char* nameSTACK;
;;; int BacktrackFlag = 0;
;;; Backtrack:
;;; for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {

  0000b 8b f1          mov esi, ecx
  0000d 8b d0          mov edx, eax
  0000f 0f be 39       movsx edi, BYTE PTR [ecx]
  00012 85 ff          test edi, edi
  00014 74 2b          je .B5.9 ; Prob 10%

.B5.2:
  00016 89 1c 24       mov DWORD PTR [esp], ebx
  00019 33 ff          xor edi, edi
  0001b 8b d8          mov ebx, eax
```

```
.B5.3:

;;;    if (*maskSTACK == '*') {

  0001d 0f be 02        movsx eax, BYTE PTR [edx]
  00020 83 f8 2a        cmp eax, 42
  00023 74 4b           je .B5.16 ; Prob 16%

.B5.4:

;;;          mask = maskSTACK+1;
;;;          if (!*mask) return 1;
;;;          name = nameSTACK;
;;;          BacktrackFlag = -1;
;;;          goto Backtrack;
;;;    }
;;;    //else if (*maskSTACK == '?') {
;;;    //} else {
;;;    else if (*maskSTACK != '?') {

  00025 83 f8 3f        cmp eax, 63
  00028 74 3b           je .B5.14 ; Prob 20%

.B5.5:

;;;          //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
;;;          if (*nameSTACK != *maskSTACK) {

  0002a 3a 06           cmp al, BYTE PTR [esi]
  0002c 74 37           je .B5.14 ; Prob 20%

.B5.6:

;;;                      if (!BacktrackFlag) return 0;

  0002e 85 ff           test edi, edi
  00030 74 5b           je .B5.20 ; Prob 4%

.B5.7:

;;;                      name++;

  00032 41              inc ecx
  00033 8b d3           mov edx, ebx
  00035 8b f1           mov esi, ecx
  00037 0f be 01        movsx eax, BYTE PTR [ecx]
  0003a 85 c0           test eax, eax
  0003c 75 df           jne .B5.3 ; Prob 82%

.B5.8:
  0003e 8b 1c 24        mov ebx, DWORD PTR [esp]

.B5.9:

;;;                      goto Backtrack;
;;;          }
;;;    }
;;; }
;;; while (*maskSTACK == '*') ++maskSTACK;

  00041 0f be 02        movsx eax, BYTE PTR [edx]
  00044 83 f8 2a        cmp eax, 42
  00047 75 09           jne .B5.13 ; Prob 37%

.B5.11:
  00049 42              inc edx
  0004a 0f be 02        movsx eax, BYTE PTR [edx]
  0004d 83 f8 2a        cmp eax, 42
  00050 74 f7           je .B5.11 ; Prob 82%

.B5.13:

;;; return (!*maskSTACK);

  00052 ba 01 00 00 00  mov edx, 1
  00057 85 c0           test eax, eax
  00059 b8 00 00 00 00  mov eax, 0
  0005e 0f 44 c2        cmove eax, edx
  00061 59              pop ecx
  00062 5f              pop edi
  00063 5e              pop esi
  00064 c3              ret

.B5.14:
  00065 46              inc esi
  00066 42              inc edx
  00067 0f be 06        movsx eax, BYTE PTR [esi]
  0006a 85 c0           test eax, eax
  0006c 75 af           jne .B5.3 ; Prob 82%
```

```
  0006e eb ce           jmp .B5.8 ; Prob 100%

.B5.16:
  00070 8d 5a 01        lea ebx, DWORD PTR [1+edx]
  00073 0f be 52 01     movsx edx, BYTE PTR [1+edx]
  00077 85 d2           test edx, edx
  00079 74 1b           je .B5.21 ; Prob 4%

.B5.17:
  0007b 0f be 3e        movsx edi, BYTE PTR [esi]
  0007e 8b ce           mov ecx, esi
  00080 8b d3           mov edx, ebx
  00082 85 ff           test edi, edi
  00084 74 b8           je .B5.8 ; Prob 18%

.B5.18:
  00086 bf ff ff ff ff  mov edi, -1
  0008b eb 90           jmp .B5.3 ; Prob 100%

.B5.20:
  0008d 8b 1c 24        mov ebx, DWORD PTR [esp]
  00090 33 c0           xor eax, eax
  00092 59              pop ecx
  00093 5f              pop edi
  00094 5e              pop esi
  00095 c3              ret

.B5.21:
  00096 8b 1c 24        mov ebx, DWORD PTR [esp]
  00099 b8 01 00 00 00  mov eax, 1
  0009e 59              pop ecx
  0009f 5f              pop edi
  000a0 5e              pop esi
  000a1 c3              ret
  000a2 8d b4 26 00 00
        00 00 8d bc 27
        00 00 00 00     ALIGN     16

_WildcardMatch_Iterative_Kaze ENDP
*/



// Igor Pavlov's recursive variant modified (and converted to iterative) by Kaze, 2013-Nov-28:
//static boolean Wildcard_IP(const char *mask, int maskPos, const char *name, int namePos)
//{
//int maskLen = maskGLOBALlen - maskPos;
//int nameLen = nameGLOBALlen - namePos;
//if (maskLen == 0)
//      if (nameLen == 0)
//            return true;
//      else
//            return false;
//if (mask[maskPos] == '*') {
//      if (Wildcard_IP(mask, maskPos + 1, name, namePos))
//            return true;
//      if (nameLen == 0)
//            return false;
//      return Wildcard_IP(mask, maskPos, name, namePos + 1);
//} else if (mask[maskPos] == '?') {
//      if (nameLen == 0)
//            return false;
//      return Wildcard_IP(mask, maskPos + 1, name, namePos + 1);
//} else {
//      if (mask[maskPos] != name[namePos])
//            return false;
//      return Wildcard_IP(mask, maskPos + 1, name, namePos + 1);
//}
//}
int WildcardMatch_Iterative_Kaze1(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
      if (*maskSTACK == '&') {
            mask = maskSTACK+1;
            if (!*mask) return 1;
            name = nameSTACK;
            BacktrackFlag = -1;
            goto Backtrack;
      }
      //else if (*maskSTACK == '+') {
      //} else {
      else if (*maskSTACK != '+') {
            //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
            if (*nameSTACK != *maskSTACK) {
                        if (!BacktrackFlag) return 0;
```

```
                          name++;
                          goto Backtrack;
              }
       }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
       fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
       if (*maskSTACK == '&') {
             mask = maskSTACK+1;
             if (!*mask) return 1;
             name = nameSTACK;
             BacktrackFlag = -1;
             goto Backtrack;
       }
       //else if (*maskSTACK == '+') {
       //} else {
       else if (*maskSTACK != '+') {
             //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
             if (*nameSTACK != *maskSTACK) {
                          if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                          name++;
                          goto Backtrack;
             }
       }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
       if (*mask == '&') {
             goto Backtrack;
       }
       //else if (*mask == '+') {
       //} else {
       else if (*mask != '+') {
             if (*name != *mask) {
                          return 0;
             }
       }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
       if (*maskSTACK == '&') {
             mask = maskSTACK+1;
             if (!*mask) return 1;
             name = nameSTACK;
             //BacktrackFlag = -1;
             goto Backtrack;
       }
       //else if (*maskSTACK == '+') {
       //} else {
       else if (*maskSTACK != '+') {
             //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
             if (*nameSTACK != *maskSTACK) {
                          //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                          name++;
                          goto Backtrack;
             }
       }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
```

```
}
int WildcardMatch_Iterative_Kaze2(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
              if (*nameSTACK != *maskSTACK) {
                          if (!BacktrackFlag) return 0;
                          name++;
                          goto Backtrack;
              }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
       fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
              if (*nameSTACK != *maskSTACK) {
                          if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                          name++;
                          goto Backtrack;
              }
        }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
        if (*mask == '&') {
              goto Backtrack;
        }
        //else if (*mask == '+') {
        //} else {
        else if (*mask != '+') {
              if (*name != *mask) {
                          return 0;
              }
        }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
```



```
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              //BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
              if (*nameSTACK != *maskSTACK) {
                            //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                            name++;
                            goto Backtrack;
              }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kaze3(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
              if (*nameSTACK != *maskSTACK) {
                            if (!BacktrackFlag) return 0;
                            name++;
                            goto Backtrack;
              }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
```

```
            if (*nameSTACK != *maskSTACK) {
                        if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                        name++;
                        goto Backtrack;
            }
      }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
      if (*mask == '&') {
            goto Backtrack;
      }
      //else if (*mask == '+') {
      //} else {
      else if (*mask != '+') {
            if (*name != *mask) {
                        return 0;
            }
      }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
      if (*maskSTACK == '&') {
            mask = maskSTACK+1;
            if (!*mask) return 1;
            name = nameSTACK;
            //BacktrackFlag = -1;
            goto Backtrack;
      }
      //else if (*maskSTACK == '+') {
      //} else {
      else if (*maskSTACK != '+') {
            //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
            if (*nameSTACK != *maskSTACK) {
                        //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                        name++;
                        goto Backtrack;
            }
      }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kaze4(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
      if (*maskSTACK == '&') {
            mask = maskSTACK+1;
            if (!*mask) return 1;
            name = nameSTACK;
            BacktrackFlag = -1;
            goto Backtrack;
      }
      //else if (*maskSTACK == '+') {
      //} else {
      else if (*maskSTACK != '+') {
            //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
            if (*nameSTACK != *maskSTACK) {
                        if (!BacktrackFlag) return 0;
                        name++;
                        goto Backtrack;
            }
      }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
      fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
```

```
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
	if (*maskSTACK == '&') {
		mask = maskSTACK+1;
		if (!*mask) return 1;
		name = nameSTACK;
		BacktrackFlag = -1;
		goto Backtrack;
	}
	//else if (*maskSTACK == '+') {
	//} else {
	else if (*maskSTACK != '+') {
		//if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
		if (*nameSTACK != *maskSTACK) {
				if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
				name++;
				goto Backtrack;
		}
	}
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
	if (*mask == '&') {
		goto Backtrack;
	}
	//else if (*mask == '+') {
	//} else {
	else if (*mask != '+') {
		if (*name != *mask) {
				return 0;
		}
	}
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
	if (*maskSTACK == '&') {
		mask = maskSTACK+1;
		if (!*mask) return 1;
		name = nameSTACK;
		//BacktrackFlag = -1;
		goto Backtrack;
	}
	//else if (*maskSTACK == '+') {
	//} else {
	else if (*maskSTACK != '+') {
		//if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
		if (*nameSTACK != *maskSTACK) {
				//if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
				name++;
				goto Backtrack;
		}
	}
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kaze5(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
	if (*maskSTACK == '&') {
		mask = maskSTACK+1;
		if (!*mask) return 1;
		name = nameSTACK;
		BacktrackFlag = -1;
		goto Backtrack;
	}
	//else if (*maskSTACK == '+') {
	//} else {
	else if (*maskSTACK != '+') {
		//if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
		if (*nameSTACK != *maskSTACK) {
				if (!BacktrackFlag) return 0;
				name++;
```

```
                        goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
       fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                        if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                        name++;
                        goto Backtrack;
                }
        }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
        if (*mask == '&') {
                goto Backtrack;
        }
        //else if (*mask == '+') {
        //} else {
        else if (*mask != '+') {
                if (*name != *mask) {
                        return 0;
                }
        }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                //BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                        //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                        name++;
                        goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
```

```
int WildcardMatch_Iterative_Kaze6(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
              if (*nameSTACK != *maskSTACK) {
                           if (!BacktrackFlag) return 0;
                           name++;
                           goto Backtrack;
              }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
       fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
              if (*nameSTACK != *maskSTACK) {
                           if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                           name++;
                           goto Backtrack;
              }
        }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
        if (*mask == '&') {
              goto Backtrack;
        }
        //else if (*mask == '+') {
        //} else {
        else if (*mask != '+') {
              if (*name != *mask) {
                           return 0;
              }
        }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
```

```
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                //BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                                name++;
                                goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kaze7(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                if (!BacktrackFlag) return 0;
                                name++;
                                goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
        fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
```

```
                    if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                    name++;
                    goto Backtrack;
          }
     }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
     if (*mask == '&') {
          goto Backtrack;
     }
     //else if (*mask == '+') {
     //} else {
     else if (*mask != '+') {
          if (*name != *mask) {
                    return 0;
          }
     }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
     if (*maskSTACK == '&') {
          mask = maskSTACK+1;
          if (!*mask) return 1;
          name = nameSTACK;
          //BacktrackFlag = -1;
          goto Backtrack;
     }
     //else if (*maskSTACK == '+') {
     //} else {
     else if (*maskSTACK != '+') {
          //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
          if (*nameSTACK != *maskSTACK) {
                    //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                    name++;
                    goto Backtrack;
          }
     }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kaze8(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
     if (*maskSTACK == '&') {
          mask = maskSTACK+1;
          if (!*mask) return 1;
          name = nameSTACK;
          BacktrackFlag = -1;
          goto Backtrack;
     }
     //else if (*maskSTACK == '+') {
     //} else {
     else if (*maskSTACK != '+') {
          //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
          if (*nameSTACK != *maskSTACK) {
                    if (!BacktrackFlag) return 0;
                    name++;
                    goto Backtrack;
          }
     }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
   // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
   int i;
   unsigned char maskOrder1[256];
   unsigned char nameOrder1[256];
   for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
   for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
   for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
   // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
     fails.
   for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
```

```
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
    if (*maskSTACK == '&') {
        mask = maskSTACK+1;
        if (!*mask) return 1;
        name = nameSTACK;
        BacktrackFlag = -1;
        goto Backtrack;
    }
    //else if (*maskSTACK == '+') {
    //} else {
    else if (*maskSTACK != '+') {
        //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
        if (*nameSTACK != *maskSTACK) {
                if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                name++;
                goto Backtrack;
        }
    }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
    if (*mask == '&') {
        goto Backtrack;
    }
    //else if (*mask == '+') {
    //} else {
    else if (*mask != '+') {
        if (*name != *mask) {
                return 0;
        }
    }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
    if (*maskSTACK == '&') {
        mask = maskSTACK+1;
        if (!*mask) return 1;
        name = nameSTACK;
        //BacktrackFlag = -1;
        goto Backtrack;
    }
    //else if (*maskSTACK == '+') {
    //} else {
    else if (*maskSTACK != '+') {
        //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
        if (*nameSTACK != *maskSTACK) {
                //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                name++;
                goto Backtrack;
        }
    }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kaze9(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
    if (*maskSTACK == '&') {
        mask = maskSTACK+1;
        if (!*mask) return 1;
        name = nameSTACK;
        BacktrackFlag = -1;
        goto Backtrack;
    }
    //else if (*maskSTACK == '+') {
    //} else {
    else if (*maskSTACK != '+') {
        //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
        if (*nameSTACK != *maskSTACK) {
                if (!BacktrackFlag) return 0;
                name++;
                goto Backtrack;
```

```
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                                name++;
                                goto Backtrack;
                }
        }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
        if (*mask == '&') {
                goto Backtrack;
        }
        //else if (*mask == '+') {
        //} else {
        else if (*mask != '+') {
                if (*name != *mask) {
                                return 0;
                }
        }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                //BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                                name++;
                                goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kaze0(const char* mask, const char* name) {
```

```
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
       if (*maskSTACK == '&') {
             mask = maskSTACK+1;
             if (!*mask) return 1;
             name = nameSTACK;
             BacktrackFlag = -1;
             goto Backtrack;
       }
       //else if (*maskSTACK == '+') {
       //} else {
       else if (*maskSTACK != '+') {
             //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
             if (*nameSTACK != *maskSTACK) {
                          if (!BacktrackFlag) return 0;
                          name++;
                          goto Backtrack;
             }
       }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
       fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
       if (*maskSTACK == '&') {
             mask = maskSTACK+1;
             if (!*mask) return 1;
             name = nameSTACK;
             BacktrackFlag = -1;
             goto Backtrack;
       }
       //else if (*maskSTACK == '+') {
       //} else {
       else if (*maskSTACK != '+') {
             //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
             if (*nameSTACK != *maskSTACK) {
                          if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                          name++;
                          goto Backtrack;
             }
       }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
       if (*mask == '&') {
             goto Backtrack;
       }
       //else if (*mask == '+') {
       //} else {
       else if (*mask != '+') {
             if (*name != *mask) {
                          return 0;
             }
       }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
```

```
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                //BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                                name++;
                                goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kazea(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                if (!BacktrackFlag) return 0;
                                name++;
                                goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
```

```
                name++;
                goto Backtrack;
            }
        }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
    if (*mask == '&') {
        goto Backtrack;
    }
    //else if (*mask == '+') {
    //} else {
    else if (*mask != '+') {
        if (*name != *mask) {
                return 0;
        }
    }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
    if (*maskSTACK == '&') {
        mask = maskSTACK+1;
        if (!*mask) return 1;
        name = nameSTACK;
        //BacktrackFlag = -1;
        goto Backtrack;
    }
    //else if (*maskSTACK == '+') {
    //} else {
    else if (*maskSTACK != '+') {
        //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
        if (*nameSTACK != *maskSTACK) {
                //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                name++;
                goto Backtrack;
        }
    }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kazeb(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
    if (*maskSTACK == '&') {
        mask = maskSTACK+1;
        if (!*mask) return 1;
        name = nameSTACK;
        BacktrackFlag = -1;
        goto Backtrack;
    }
    //else if (*maskSTACK == '+') {
    //} else {
    else if (*maskSTACK != '+') {
        //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
        if (*nameSTACK != *maskSTACK) {
                if (!BacktrackFlag) return 0;
                name++;
                goto Backtrack;
        }
    }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
  // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
  int i;
  unsigned char maskOrder1[256];
  unsigned char nameOrder1[256];
  for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
  for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
  for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
  // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
    fails.
  for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
```

```
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                                name++;
                                goto Backtrack;
                }
        }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
        if (*mask == '&') {
                goto Backtrack;
        }
        //else if (*mask == '+') {
        //} else {
        else if (*mask != '+') {
                if (*name != *mask) {
                                return 0;
                }
        }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                //BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                                name++;
                                goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kazec(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                if (!BacktrackFlag) return 0;
                                name++;
                                goto Backtrack;
                }
```

```
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                                name++;
                                goto Backtrack;
                }
        }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
        if (*mask == '&') {
                goto Backtrack;
        }
        //else if (*mask == '+') {
        //} else {
        else if (*mask != '+') {
                if (*name != *mask) {
                                return 0;
                }
        }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
                mask = maskSTACK+1;
                if (!*mask) return 1;
                name = nameSTACK;
                //BacktrackFlag = -1;
                goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
                //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
                if (*nameSTACK != *maskSTACK) {
                                //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                                name++;
                                goto Backtrack;
                }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kazed(const char* mask, const char* name) {
// Revision 1:
```

```
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
              if (*nameSTACK != *maskSTACK) {
                            if (!BacktrackFlag) return 0;
                            name++;
                            goto Backtrack;
              }
        }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
       fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
              mask = maskSTACK+1;
              if (!*mask) return 1;
              name = nameSTACK;
              BacktrackFlag = -1;
              goto Backtrack;
        }
        //else if (*maskSTACK == '+') {
        //} else {
        else if (*maskSTACK != '+') {
              //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
              if (*nameSTACK != *maskSTACK) {
                            if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                            name++;
                            goto Backtrack;
              }
        }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
        if (*mask == '&') {
              goto Backtrack;
        }
        //else if (*mask == '+') {
        //} else {
        else if (*mask != '+') {
              if (*name != *mask) {
                            return 0;
              }
        }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
        if (*maskSTACK == '&') {
```

```
        mask = maskSTACK+1;
        if (!*mask) return 1;
        name = nameSTACK;
        //BacktrackFlag = -1;
        goto Backtrack;
    }
    //else if (*maskSTACK == '+') {
    //} else {
    else if (*maskSTACK != '+') {
        //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
        if (*nameSTACK != *maskSTACK) {
                //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                name++;
                goto Backtrack;
        }
    }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kazee(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
    if (*maskSTACK == '&') {
        mask = maskSTACK+1;
        if (!*mask) return 1;
        name = nameSTACK;
        BacktrackFlag = -1;
        goto Backtrack;
    }
    //else if (*maskSTACK == '+') {
    //} else {
    else if (*maskSTACK != '+') {
        //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
        if (*nameSTACK != *maskSTACK) {
                if (!BacktrackFlag) return 0;
                name++;
                goto Backtrack;
        }
    }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
   // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
   int i;
   unsigned char maskOrder1[256];
   unsigned char nameOrder1[256];
   for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
   for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
   for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
   // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
     fails.
   for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
   int i;
   for (i=0; i < strlen(mask); i++) {
      if ( mask[i] != '&' && mask[i] != '+' )
         if ( !memchr(name,mask[i],strlen(name)) ) return 0;
   }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
    if (*maskSTACK == '&') {
        mask = maskSTACK+1;
        if (!*mask) return 1;
        name = nameSTACK;
        BacktrackFlag = -1;
        goto Backtrack;
    }
    //else if (*maskSTACK == '+') {
    //} else {
    else if (*maskSTACK != '+') {
        //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
        if (*nameSTACK != *maskSTACK) {
                if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                name++;
```

```
			goto Backtrack;
		}
	}
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
	if (*mask == '&') {
		goto Backtrack;
	}
	//else if (*mask == '+') {
	//} else {
	else if (*mask != '+') {
		if (*name != *mask) {
			return 0;
		}
	}
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
	if (*maskSTACK == '&') {
		mask = maskSTACK+1;
		if (!*mask) return 1;
		name = nameSTACK;
		//BacktrackFlag = -1;
		goto Backtrack;
	}
	//else if (*maskSTACK == '+') {
	//} else {
	else if (*maskSTACK != '+') {
		//if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
		if (*nameSTACK != *maskSTACK) {
			//if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
			name++;
			goto Backtrack;
		}
	}
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}
int WildcardMatch_Iterative_Kazef(const char* mask, const char* name) {
// Revision 1:
/*
const char* maskSTACK;
const char* nameSTACK;
int BacktrackFlag = 0;
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
	if (*maskSTACK == '&') {
		mask = maskSTACK+1;
		if (!*mask) return 1;
		name = nameSTACK;
		BacktrackFlag = -1;
		goto Backtrack;
	}
	//else if (*maskSTACK == '+') {
	//} else {
	else if (*maskSTACK != '+') {
		//if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
		if (*nameSTACK != *maskSTACK) {
			if (!BacktrackFlag) return 0;
			name++;
			goto Backtrack;
		}
	}
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
*/
// Revision 2, 2013-Nov-30:
const char* maskSTACK;
const char* nameSTACK;
//int BacktrackFlag = 0; // No need of it in rev.2
/*
    // Simplest heuristic with SUPEROVERHEAD enforced: trying to skip the whole wildcard section by comparing the two arrays order 1 of mask&name.
    int i;
    unsigned char maskOrder1[256];
    unsigned char nameOrder1[256];
    for (i='a'; i <= 'z'; i++) { maskOrder1[i]=0; nameOrder1[i]=0;}
    for (i=0; i < strlen(mask); i++) maskOrder1[mask[i]]=1;
    for (i=0; i < strlen(name); i++) nameOrder1[name[i]]=1;
    // Assuming the incoming strings are already lowercased (as it should for speed) and if we don't have matching alphabet parts (from mask side) means we don't need to compare any further i.e. the match
      fails.
    for (i='a'; i <= 'z'; i++) if ( maskOrder1[i] == 1 && nameOrder1[i] == 0 ) return 0;
*/
/*
    int i;
```

```
    for (i=0; i < strlen(mask); i++) {
        if ( mask[i] != '&' && mask[i] != '+' )
            if ( !memchr(name,mask[i],strlen(name)) ) return 0;
    }
*/
/*
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
       if (*maskSTACK == '&') {
             mask = maskSTACK+1;
             if (!*mask) return 1;
             name = nameSTACK;
             BacktrackFlag = -1;
             goto Backtrack;
       }
       //else if (*maskSTACK == '+') {
       //} else {
       else if (*maskSTACK != '+') {
             //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
             if (*nameSTACK != *maskSTACK) {
                          if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                          name++;
                          goto Backtrack;
             }
       }
}
*/
// Here, outside the main/second 'for', in order to avoid branching we need to set the old/obsolete BacktrackFlag i.e. we need first occurrence of '&':
for (name, mask; *name; ++name, ++mask) {
       if (*mask == '&') {
             goto Backtrack;
       }
       //else if (*mask == '+') {
       //} else {
       else if (*mask != '+') {
             if (*name != *mask) {
                          return 0;
             }
       }
}
// We are entering the main/second 'for' with mask pointing to '&' as if BacktrackFlag is already set in the very first iteration at first condition:
Backtrack:
for (nameSTACK = name, maskSTACK = mask; *nameSTACK; ++nameSTACK, ++maskSTACK) {
       if (*maskSTACK == '&') {
             mask = maskSTACK+1;
             if (!*mask) return 1;
             name = nameSTACK;
             //BacktrackFlag = -1;
             goto Backtrack;
       }
       //else if (*maskSTACK == '+') {
       //} else {
       else if (*maskSTACK != '+') {
             //if (tolower(*nameSTACK) != tolower(*maskSTACK)) { // These 'tolower's are outrageous, they hurt speed BADLY, in real-world usage both should have been lowercased outwith the 'for'.
             if (*nameSTACK != *maskSTACK) {
                          //if (!BacktrackFlag) return 0; // Stupid branching, SLOW!
                          name++;
                          goto Backtrack;
             }
       }
}
while (*maskSTACK == '&') ++maskSTACK;
return (!*maskSTACK);
}


static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_1(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen1 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
```

```
      return true;                                                    //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
```

```
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') ¦¦ (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_2(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen2 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                        //         uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;
```

```
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos + 1, name, namePos + 1);
  }
//      wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, maskPos, name, namePos + 1);
```

```
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_3(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen3 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                  //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
```

```
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_4(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen4 - namePos;
```

```
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                   //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '|') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos + 1, name, namePos + 1);
  }
```

```
  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos, name, namePos + 1);
  }



  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_5(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen5 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                 //        uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;
    */
```

```
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '|') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;
```

```
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_6(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen6 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                  //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;
    */
```

```
    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
```

```
            }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_7(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen7 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') ¦¦ (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
```

```
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_8(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen8 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
```

```
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                     //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos + 1, name, namePos + 1);
  }
//      wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '|') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos + 1, name, namePos + 1);
  }
//      wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;
```

```
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') ¦¦ (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos, name, namePos + 1);
  }



  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
              if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
              } else {
      if ((maskChar) != (c))
        return false;
              }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_9(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen9 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                  //        uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos + 1, name, namePos + 1);
  }
//        wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;
```

```
    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos + 1, name, namePos + 1);
  }
//      wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos + 1, name, namePos))
      return true;
```

```
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_0(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlen0 - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                      //        uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos + 1, name, namePos + 1);
  }
//        wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos + 1, name, namePos + 1);
  }
//        wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
```

```
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_a(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
```

```
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlena - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
```

```
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_b(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlenb - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                  //      uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
```

```
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '¦'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '¦') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') ¦¦ (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') ¦¦ (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;
```

```
    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_c(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlenc - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                    //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '|') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
```

```
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos + 1, name, namePos + 1);
  }
//      wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
```

```
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_d(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlend - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == ’@’) // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether ’or empty’ or ’and not empty’
      return true;                                                //       uncommented is ’or empty’
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == ’#’) // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard ’¦’/’~’ any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == ’¦’) // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= ’A’) && (KAZE_toupper(c) <= ’Z’) ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == ’~’) // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= ’A’) && (KAZE_toupper(c) <= ’Z’) ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard ’^’/’$’ any ALPHA character {or empty}/{and not empty},
  else if(maskChar == ’^’) // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < ’A’) ¦¦ (KAZE_toupper(c) > ’Z’) ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
```

```
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_e(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlene - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
```

```
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                  //       uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos + 1, name, namePos + 1);
  }
//      wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '|') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos + 1, name, namePos + 1);
  }
//      wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;
```

```
    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos, name, namePos + 1);
  }



  else if(maskChar == '*')
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask,  maskPos + 1, name, namePos + 1);
  }
}

static boolean EnhancedMaskTest_OrEmpty_AndNotEmpty_f(const char *mask, int maskPos, const char *name, int namePos)
{
  char maskChar;
  char c;
  //int maskLen = KAZE_strlen(mask) - maskPos;
  //int nameLen = KAZE_strlenLF(name) - namePos;
  // Above 2 lines are modified with GLOBAL variables for speed as follows:
  int maskLen = maskGLOBALlen - maskPos; // for speed up ((12415s - 755s)/755)*100% = 1544%
  int nameLen = nameGLOBALlenf - namePos;
  if (maskLen == 0) {
    if (nameLen == 0)
      return true;
    else
      return false;
  }
  maskChar = mask[maskPos];
  if (maskChar == '@') // or empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos)) // KAZE: THIS LINE DECIDES whether 'or empty' or 'and not empty'
      return true;                                                  //        uncommented is 'or empty'
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '#') // and not empty
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;
    */
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos + 1, name, namePos + 1);
  }
//        wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty},
  else if(maskChar == '|') // or empty AND NOT ALPHA
  {
```

```
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos + 1, name, namePos + 1);
  }
//       wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
  else if(maskChar == '^') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos + 1, name, namePos + 1);
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
    /*
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;
    */

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos + 1, name, namePos + 1);
  }


  else if(maskChar == '%') // or empty AND NOT ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) >= 'A') && (KAZE_toupper(c) <= 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos, name, namePos + 1);
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {

    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;

    c = name[namePos];
      if ( (KAZE_toupper(c) < 'A') || (KAZE_toupper(c) > 'Z') ) // Stupidly slow: make it faster ...
        return false;

    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos, name, namePos + 1);
  }


  else if(maskChar == '*')
```

```
  {
    if (EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos + 1, name, namePos))
      return true;
    if (nameLen == 0)
      return false;
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, maskPos, name, namePos + 1);
  }
  else
  {
    c = name[namePos];
    //if (maskChar != c)
             if (CaseSensitiveWildcardMatching_flag == 0) {
      if (KAZE_toupper(maskChar) != KAZE_toupper(c))
        return false;
             } else {
      if ((maskChar) != (c))
        return false;
             }
    return EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask,  maskPos + 1, name, namePos + 1);
  }
}



/*
boolean CompareWildCardWithName1(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen1 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_1(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithName2(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen2 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_2(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithName3(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen3 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_3(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithName4(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen4 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_4(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithName5(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen5 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_5(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
```

```
boolean CompareWildCardWithName6(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen6 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_6(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithName7(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen7 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_7(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithName8(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen8 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_8(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithName9(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen9 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_9(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithName0(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlen0 = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_0(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithNamea(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlena = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_a(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithNameb(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlenb = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_b(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
```

```
  return Txpbool;
}
boolean CompareWildCardWithNamec(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlenc = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_c(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithNamed(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlend = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_d(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithNamee(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlene = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_e(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
boolean CompareWildCardWithNamef(const char *mask, const char *name)
{
  boolean Txpbool;
  //VIVA_IgorPavlov_invocations_global_counter++;

  // The two lines below are nasty, Very slow:
  //maskGLOBALlen = KAZE_strlen(mask);
  //nameGLOBALlenf = KAZE_strlen(name);

  Txpbool = EnhancedMaskTest_OrEmpty_AndNotEmpty_f(mask, 0, name, 0);
  //if (Txpbool) WildGLOBALhits++;
  return Txpbool;
}
*/

// ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
// Modified Igor Pavlov's 7zip wildcard code ]

/*
                          ...

                        if( wrdlen < MaxLineLength +1+1)
                        { wrd[ wrdlen ] = KAZE_tolower( workbyte ); }

                        if (workbyte == 10) {NumberOfLines++;
                            if (wrdlen > LongestLine) LongestLine=wrdlen;
                            if (wrdlen < ShortestLine) ShortestLine=wrdlen;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MaxLineLength +1+1)
                          {
                            wrd[ wrdlen ] = 0;
                                        if ( CompareWildCardWithName ( LINE10_2, wrd ) ) {
                                                    //printf("%s\n", wrd);
                                                    fprintf( fp_outLOG, "%s\n", wrd); DumpedLines++;
                                        if ((DumpedLines & 0xff) == 0xff)
                                                    //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                                    fflush(fp_outLOG); // Not sure: CTRL+C doesn't flush?!
                                        }

                          }
// Wildcard search ]
                            wrdlen = 0;
                        }
                        else wrdlen++;

                          ...
*/
```

```
// SLOW, recursive:
//      wildcard '*' any character(s) or empty,
//      wildcard '.' any ALPHA character(s) or empty,
//      wildcard '%' any NON-ALPHA character(s) or empty,
//      wildcard '@'/'#' any character {or empty}/{and not empty},
//      wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
//      wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty},
//
// SLOW mode is clearly written, however recursive calls are very "overheadY" also it has scary branching:
/*
  if (maskChar == '@') // or empty
  {
  }
  else if(maskChar == '#') // and not empty
  {
  }
  else if(maskChar == '|') // or empty AND NOT ALPHA
  {
  }
  else if(maskChar == '~') // and not empty AND NOT ALPHA
  {
  }
  else if(maskChar == '^') // or empty AND ALPHA
  {
  }
  else if(maskChar == '$') // and not empty AND ALPHA
  {
  }
  else if(maskChar == '%') // or empty AND NOT ALPHA
  {
  }
  else if(maskChar == '.') // or empty AND ALPHA
  {
  }
  else if(maskChar == '*')
  {
  }
  else
  {
  }
*/
// The way it is done (in Kazahana) now the SLOW overrides the FAST, meaning that presence of at least one of the 9 wildcards trumps the FAST mode.
// FAST, iterative:
//      wildcard '&' any character(s) or empty,
//      wildcard '+' any character and not empty.
// CAUTION: For speed case-insensitivity is achieved outwith the function i.e. by default it is case-sensitive unless you did take care of the opposite.
static boolean IterativeWildcards1(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
       if ( (*mask != *name) && (*mask != '+') )
             return false;
       mask++;
       name++;
}
while ( *name ) {
       if ( *mask == '&' ) {
             if (!*++mask)
                          return true;
             maskSTACK = mask;
             nameSTACK = name + 1;
       } else if ( (*mask == *name) || (*mask == '+') ) {
             mask++;
             name++;
       } else {
             mask = maskSTACK;
             name = nameSTACK++;
       }
}
while ( *mask == '&' )
       mask++;
return !*mask;
}
static boolean IterativeWildcards2(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
       if ( (*mask != *name) && (*mask != '+') )
             return false;
       mask++;
       name++;
}
while ( *name ) {
       if ( *mask == '&' ) {
             if (!*++mask)
```

```
                                return true;
                maskSTACK = mask;
                nameSTACK = name + 1;
        } else if ( (*mask == *name) || (*mask == '+') ) {
                mask++;
                name++;
        } else {
                mask = maskSTACK;
                name = nameSTACK++;
        }
}
while ( *mask == '&' )
        mask++;
return !*mask;
}
static boolean IterativeWildcards3(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
        if ( (*mask != *name) && (*mask != '+') )
                return false;
        mask++;
        name++;
}
while ( *name ) {
        if ( *mask == '&' ) {
                if (!*++mask)
                                return true;
                maskSTACK = mask;
                nameSTACK = name + 1;
        } else if ( (*mask == *name) || (*mask == '+') ) {
                mask++;
                name++;
        } else {
                mask = maskSTACK;
                name = nameSTACK++;
        }
}
while ( *mask == '&' )
        mask++;
return !*mask;
}
static boolean IterativeWildcards4(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
        if ( (*mask != *name) && (*mask != '+') )
                return false;
        mask++;
        name++;
}
while ( *name ) {
        if ( *mask == '&' ) {
                if (!*++mask)
                                return true;
                maskSTACK = mask;
                nameSTACK = name + 1;
        } else if ( (*mask == *name) || (*mask == '+') ) {
                mask++;
                name++;
        } else {
                mask = maskSTACK;
                name = nameSTACK++;
        }
}
while ( *mask == '&' )
        mask++;
return !*mask;
}
static boolean IterativeWildcards5(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
        if ( (*mask != *name) && (*mask != '+') )
                return false;
        mask++;
        name++;
}
while ( *name ) {
        if ( *mask == '&' ) {
                if (!*++mask)
                                return true;
                maskSTACK = mask;
```

```
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
        mask++;
        name++;
    } else {
        mask = maskSTACK;
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}
static boolean IterativeWildcards6(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
    if ( (*mask != *name) && (*mask != '+') )
        return false;
    mask++;
    name++;
}
while ( *name ) {
    if ( *mask == '&' ) {
        if (!*++mask)
                return true;
        maskSTACK = mask;
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
        mask++;
        name++;
    } else {
        mask = maskSTACK;
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}
static boolean IterativeWildcards7(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
    if ( (*mask != *name) && (*mask != '+') )
        return false;
    mask++;
    name++;
}
while ( *name ) {
    if ( *mask == '&' ) {
        if (!*++mask)
                return true;
        maskSTACK = mask;
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
        mask++;
        name++;
    } else {
        mask = maskSTACK;
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}
static boolean IterativeWildcards8(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
    if ( (*mask != *name) && (*mask != '+') )
        return false;
    mask++;
    name++;
}
while ( *name ) {
    if ( *mask == '&' ) {
        if (!*++mask)
                return true;
        maskSTACK = mask;
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
```

```
            mask++;
            name++;
        } else {
            mask = maskSTACK;
            name = nameSTACK++;
        }
}
while ( *mask == '&' )
        mask++;
return !*mask;
}
static boolean IterativeWildcards9(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
        if ( (*mask != *name) && (*mask != '+') )
            return false;
        mask++;
        name++;
}
while ( *name ) {
        if ( *mask == '&' ) {
            if (!*++mask)
                        return true;
            maskSTACK = mask;
            nameSTACK = name + 1;
        } else if ( (*mask == *name) || (*mask == '+') ) {
            mask++;
            name++;
        } else {
            mask = maskSTACK;
            name = nameSTACK++;
        }
}
while ( *mask == '&' )
        mask++;
return !*mask;
}
static boolean IterativeWildcards0(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
        if ( (*mask != *name) && (*mask != '+') )
            return false;
        mask++;
        name++;
}
while ( *name ) {
        if ( *mask == '&' ) {
            if (!*++mask)
                        return true;
            maskSTACK = mask;
            nameSTACK = name + 1;
        } else if ( (*mask == *name) || (*mask == '+') ) {
            mask++;
            name++;
        } else {
            mask = maskSTACK;
            name = nameSTACK++;
        }
}
while ( *mask == '&' )
        mask++;
return !*mask;
}
static boolean IterativeWildcardsa(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
        if ( (*mask != *name) && (*mask != '+') )
            return false;
        mask++;
        name++;
}
while ( *name ) {
        if ( *mask == '&' ) {
            if (!*++mask)
                        return true;
            maskSTACK = mask;
            nameSTACK = name + 1;
        } else if ( (*mask == *name) || (*mask == '+') ) {
            mask++;
            name++;
```

```
    } else {
        mask = maskSTACK;
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}
static boolean IterativeWildcardsb(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
    if ( (*mask != *name) && (*mask != '+') )
        return false;
    mask++;
    name++;
}
while ( *name ) {
    if ( *mask == '&' ) {
        if (!*++mask)
                return true;
        maskSTACK = mask;
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
        mask++;
        name++;
    } else {
        mask = maskSTACK;
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}
static boolean IterativeWildcardsc(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
    if ( (*mask != *name) && (*mask != '+') )
        return false;
    mask++;
    name++;
}
while ( *name ) {
    if ( *mask == '&' ) {
        if (!*++mask)
                return true;
        maskSTACK = mask;
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
        mask++;
        name++;
    } else {
        mask = maskSTACK;
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}
static boolean IterativeWildcardsd(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
    if ( (*mask != *name) && (*mask != '+') )
        return false;
    mask++;
    name++;
}
while ( *name ) {
    if ( *mask == '&' ) {
        if (!*++mask)
                return true;
        maskSTACK = mask;
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
        mask++;
        name++;
    } else {
        mask = maskSTACK;
```

```
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}
static boolean IterativeWildcardse(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
    if ( (*mask != *name) && (*mask != '+') )
        return false;
    mask++;
    name++;
}
while ( *name ) {
    if ( *mask == '&' ) {
        if (!*++mask)
                return true;
        maskSTACK = mask;
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
        mask++;
        name++;
    } else {
        mask = maskSTACK;
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}
static boolean IterativeWildcardsf(const char *mask, const char *name)
{
const char *maskSTACK;
const char *nameSTACK;

while ( (*name) && (*mask != '&') ) {
    if ( (*mask != *name) && (*mask != '+') )
        return false;
    mask++;
    name++;
}
while ( *name ) {
    if ( *mask == '&' ) {
        if (!*++mask)
                return true;
        maskSTACK = mask;
        nameSTACK = name + 1;
    } else if ( (*mask == *name) || (*mask == '+') ) {
        mask++;
        name++;
    } else {
        mask = maskSTACK;
        name = nameSTACK++;
    }
}
while ( *mask == '&' )
    mask++;
return !*mask;
}


//#define _WIN32ASM_

#if defined(_WIN32ASM_)
// Optimizing subroutines in assembly language
// An optimization guide for x86 platforms
// By Agner Fog. Copenhagen University College of Engineering.
/*
It is possible to calculate the absolute value of a signed integer without branching:
; Example 9.15, Calculate absolute value of eax
    cdq ; Copy sign bit of eax to all bits of edx
    xor eax, edx ; Invert all bits if negative
    sub eax, edx ; Add 1 if negative
The following example finds the minimum of two unsigned numbers: if (b > a) b = a;
; Example 9.16a, Find minimum of eax and ebx (unsigned):
    sub eax, ebx ; = a-b
    sbb edx, edx ; = (b > a) ? 0xFFFFFFFF : 0
    and edx, eax ; = (b > a) ? a-b : 0
    add ebx, edx ; Result is in ebx
*/
// Sadly MS 64bit compiler accepts not the inline ASM: error C4235: nonstandard extension used : '__asm' keyword not supported on this architecture
// MASM style inline assembly, 32 bit mode
unsigned int abs_AF (int n) {
```

```
__asm {
       mov eax, n          // Move n to eax
                           // abs(n) is calculated by inverting all bits and adding 1 if n < 0:
       cdq                 // Get sign bit into all bits of edx
       xor eax, edx        // Invert bits if negative
       sub eax, edx        // Add 1 if negative. Now eax = abs(n)
}                          // Return value is in eax
}
unsigned int min_AF (int a, int b, int c) {
__asm {
       mov eax, a          // Move a to eax
       mov ebx, b          // Move b to ebx

       sub eax, ebx ; = a-b
       sbb edx, edx ; = (b > a) ? 0xFFFFFFFF : 0
       and edx, eax ; = (b > a) ? a-b : 0
       add ebx, edx ; Result is in ebx

       mov eax, c          // Move c to eax

       sub eax, ebx ; = a-b
       sbb edx, edx ; = (b > a) ? 0xFFFFFFFF : 0
       and edx, eax ; = (b > a) ? a-b : 0
       add ebx, edx ; Result is in ebx

       mov eax, ebx ; Return value is in eax
}
}
#endif

void x64toaKAZE (      /* stdcall is faster and smaller... Might as well use it for the helper. */
        unsigned long long val,
        char *buf,
        unsigned radix,
        int is_neg
        )
{
        char *p;                /* pointer to traverse string */
        char *firstdig;         /* pointer to first digit */
        char temp;              /* temp char */
        unsigned digval;        /* value of digit */

        p = buf;

        if ( is_neg )
        {
            *p++ = '-';         /* negative, so output '-' and negate */
            val = (unsigned long long)(-(long long)val);
        }

        firstdig = p;           /* save pointer to first digit */

        do {
            digval = (unsigned) (val % radix);
            val /= radix;       /* get next digit */

            /* convert to ascii and store */
            if (digval > 9)
                *p++ = (char) (digval - 10 + 'a');  /* a letter */
            else
                *p++ = (char) (digval + '0');       /* a digit */
        } while (val > 0);

        /* We now have the digit of the number in the buffer, but in reverse
           order.  Thus we reverse them now. */

        *p-- = '\0';            /* terminate string; p points to last digit */

        do {
            temp = *p;
            *p = *firstdig;
            *firstdig = temp;   /* swap *p and *firstdig */
            --p;
            ++firstdig;         /* advance to next two digits */
        } while (firstdig < p); /* repeat until halfway */
}

/* Actual functions just call conversion helper with neg flag set correctly,
   and return pointer to buffer. */

char * _i64toaKAZE (
        long long val,
        char *buf,
        int radix
        )
{
        x64toaKAZE((unsigned long long)val, buf, radix, (radix == 10 && val < 0));
        return buf;
}
```

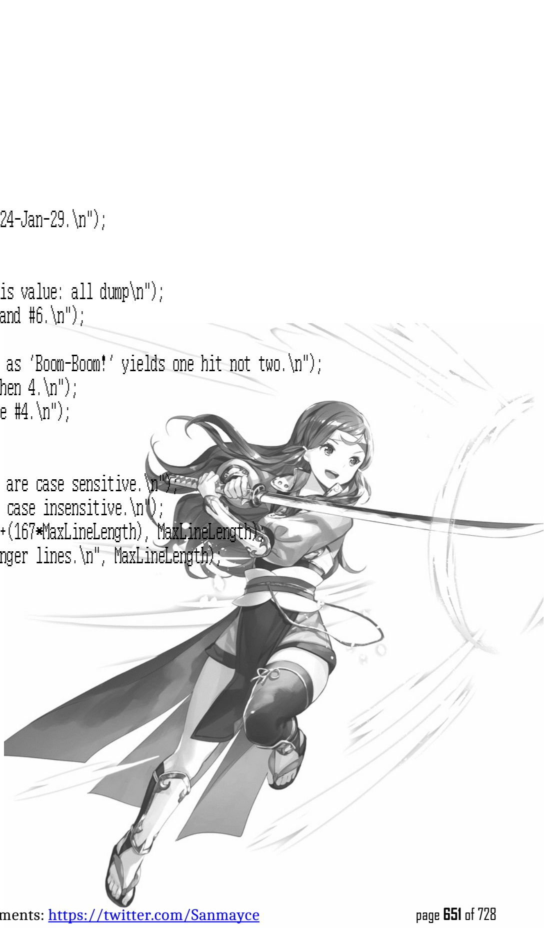
```
char * _ui64toaKAZE (
        unsigned long long val,
        char *buf,
        int radix
        )
{
        x64toaKAZE(val, buf, radix, 0);
        return buf;
}

char * _ui64toaKAZEzerocomma (
        unsigned long long val,
        char *buf,
        int radix
        )
{
                        char *p;
                        char temp;
                        int txpman;
                        int pxnman;
        x64toaKAZE(val, buf, radix, 0);
                        p = buf;
                        do {
                        } while (*++p != ’\0’);
                        p--; // p points to last digit
                             // buf points to first digit
                        buf[26] = 0;
                        txpman = 1;
                        pxnman = 0;
                        do
                        { if (buf <= p)
                          { temp = *p;
                            buf[26-txpman] = temp; pxnman++;
                            p--;
                            if (pxnman % 3 == 0)
                            { txpman++;
                              buf[26-txpman] = (char) (’,’);
                            }
                          }
                          else
                          { buf[26-txpman] = (char) (’0’); pxnman++;
                            if (pxnman % 3 == 0)
                            { txpman++;
                              buf[26-txpman] = (char) (’,’);
                            }
                          }
                          txpman++;
                        } while (txpman <= 26);
        return buf;
}

char * _ui64toaKAZEcomma (
        unsigned long long val,
        char *buf,
        int radix
        )
{
                        char *p;
                        char temp;
                        int txpman;
                        int pxnman;
        x64toaKAZE(val, buf, radix, 0);
                        p = buf;
                        do {
                        } while (*++p != ’\0’);
                        p--; // p points to last digit
                             // buf points to first digit
                        buf[26] = 0;
                        txpman = 1;
                        pxnman = 0;
                        while (buf <= p)
                        { temp = *p;
                          buf[26-txpman] = temp; pxnman++;
                          p--;
                          if (pxnman % 3 == 0 && buf <= p)
                          { txpman++;
                            buf[26-txpman] = (char) (’,’);
                          }
                          txpman++;
                        }
        return buf+26-(txpman-1);
}

// Wagner?Fischer algorithm
// From Wikipedia, the free encyclopedia
/*
 int LevenshteinDistance(char s[1..m], char t[1..n])
 {
```

```
   // for all i and j, d[i,j] will hold the Levenshtein distance between
   // the first i characters of s and the first j characters of t;
   // note that d has (m+1)x(n+1) values
   declare int d[0..m, 0..n]

   for i from 0 to m
     d[i, 0] := i // the distance of any first string to an empty second string
   for j from 0 to n
     d[0, j] := j // the distance of any second string to an empty first string

   for j from 1 to n
   {
     for i from 1 to m
     {
       if s[i] = t[j] then
         d[i, j] := d[i-1, j-1]       // no operation required
       else
         d[i, j] := minimum
                    (
                      d[i-1, j] + 1,  // a deletion
                      d[i, j-1] + 1,  // an insertion
                      d[i-1, j-1] + 1 // a substitution
                    )
     }
   }

   return d[m,n]
 }
*/

#if defined(_WIN32_ENVIRONMENT_)
#include <io.h> // needed for Windows' 'lseeki64' and 'telli64'
#else
#endif /* defined(_WIN32_ENVIRONMENT_)  */

#ifdef Commence_OpenMP
#include <omp.h>
#endif

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

        #define THREADSnumber 16
        //#define CACHEsize MasterBuffer*(1<<20) // /7*4/3
        int CACHEsize;
        #define MAXxgamsInCACHE 16*4096 // 16 threads * 4096 xgrams * 128 bytes = 8,388,608
        //#define MaxLineLength 126+100
// !!! How unexpected !!!
// I assumed the above line means 226 always yet the compilers gave:
// MaxLineLength = 226
// (167*MaxLineLength) = 21142
// MaxLineLength +(167*MaxLineLength) = 21368
// So never again such sums, they mean (167*MaxLineLength) = (167*126+100) instead of (167*(126+100))
        #define MaxLineLength 156

        // Stupid bug fixed, 'unsigned int' was an error during pasting code from Galadriel:
        int LevenshteinT1[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT2[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT3[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT4[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT5[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT6[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT7[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT8[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT9[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinT0[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinTa[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinTb[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinTc[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinTd[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinTe[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]
        int LevenshteinTf[MaxLineLength+1][MaxLineLength+1]; // declare int d[0..m, 0..n]

#define MIN(x,y) ((x) < (y) ? (x) : (y))
#define MAX(x,y) ((x) > (y) ? (x) : (y))
int main(int argc, char **argv) {

FILE *fp_inLINE;
FILE *fp_outLINE;

unsigned char workK[1024*32];
signed int workKoffset = -1;

unsigned long long FilesLEN;
unsigned long long k, k2, k3, kStartOfLine,StartOfLineFLAG;
unsigned int LINE10len, wrdlen;
```

```
#if defined(_WIN32_ENVIRONMENT_)
      unsigned long long size_inLINESIXFOUR;
#else
      //size_t size_inLINESIXFOUR;
      unsigned long long size_inLINESIXFOUR;
#endif /* defined(_WIN32_ENVIRONMENT_)  */

unsigned long long size_inLINESIXFOURleftforparsing;

// 2024-Jan-09 commented, using malloc() [
char wrdLOW[168*MaxLineLength+1+1]; // crlf
char wrdARG[168*MaxLineLength+1+1]; // crlf
char wrd[168*MaxLineLength+1+1]; // crlf
//char * wrdLOW; // crlf
//char * wrdARG; // crlf
//char * wrd; // crlf
// 2024-Jan-09 commented, using malloc() ]
char wrdCACHED[MaxLineLength+1+1]; // crlf
char workbyte;
       int xgamsInCACHE = 0;
       int CACHEremainder;
       unsigned int memory_size;
//     unsigned char xgamsCACHE[MAXxgamsInCACHE*(MaxLineLength+1+1)]; // crlf
       char * xgamsCACHE;
       char * FOUNDinPTR;
       char * xgamsCACHEMEMCHR;
       char wrdCACHEDT1[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT2[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT3[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT4[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT5[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT6[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT7[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT8[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT9[MaxLineLength+1+1]; // crlf
       char wrdCACHEDT0[MaxLineLength+1+1]; // crlf
       char wrdCACHEDTa[MaxLineLength+1+1]; // crlf

       char wrdCACHEDTb[MaxLineLength+1+1]; // crlf
       char wrdCACHEDTc[MaxLineLength+1+1]; // crlf
       char wrdCACHEDTd[MaxLineLength+1+1]; // crlf
       char wrdCACHEDTe[MaxLineLength+1+1]; // crlf
       char wrdCACHEDTf[MaxLineLength+1+1]; // crlf
time_t t1, t2, t3;
clock_t clocks1, clocks2, clocks3, clocks4;
unsigned long long TotalLines=0;
unsigned long long WordsChecked=0;
unsigned long long DumpedLines=0;
// All these needed because of pseudo-bug (see comments at the bottom) uncrushed:
unsigned long long DumpedLines1=0;
unsigned long long DumpedLines2=0;
unsigned long long DumpedLines3=0;
unsigned long long DumpedLines4=0;
unsigned long long DumpedLines5=0;
unsigned long long DumpedLines6=0;
unsigned long long DumpedLines7=0;
unsigned long long DumpedLines8=0;
unsigned long long DumpedLines9=0;
unsigned long long DumpedLines0=0;
unsigned long long DumpedLinesa=0;
unsigned long long DumpedLinesb=0;
unsigned long long DumpedLinesc=0;
unsigned long long DumpedLinesd=0;
unsigned long long DumpedLinese=0;
unsigned long long DumpedLinesf=0;

unsigned long long TotalLines1=0;
unsigned long long TotalLines2=0;
unsigned long long TotalLines3=0;
unsigned long long TotalLines4=0;
unsigned long long TotalLines5=0;

unsigned long long TotalLines6=0;
unsigned long long TotalLines7=0;
unsigned long long TotalLines8=0;
unsigned long long TotalLines9=0;
unsigned long long TotalLines0=0;
unsigned long long TotalLinesa=0;
unsigned long long TotalLinesb=0;
unsigned long long TotalLinesc=0;
unsigned long long TotalLinesd=0;
unsigned long long TotalLinese=0;
unsigned long long TotalLinesf=0;

unsigned long long WordsChecked1=0;
unsigned long long WordsChecked2=0;
unsigned long long WordsChecked3=0;
unsigned long long WordsChecked4=0;
unsigned long long WordsChecked5=0;
```

```
unsigned long long WordsChecked6=0;
unsigned long long WordsChecked7=0;
unsigned long long WordsChecked8=0;
unsigned long long WordsChecked9=0;
unsigned long long WordsChecked0=0;
unsigned long long WordsCheckeda=0;
unsigned long long WordsCheckedb=0;
unsigned long long WordsCheckedc=0;
unsigned long long WordsCheckedd=0;
unsigned long long WordsCheckede=0;
unsigned long long WordsCheckedf=0;

int AtMostLevenshteinDistance;
unsigned int SkipHeuristic;
unsigned int StartingPosition;
        unsigned int WorkAreaLedgeT1;
        unsigned int WorkAreaRedgeT1;
        unsigned int WorkAreaLedgeT2;
        unsigned int WorkAreaRedgeT2;
        unsigned int WorkAreaLedgeT3;
        unsigned int WorkAreaRedgeT3;
        unsigned int WorkAreaLedgeT4;
        unsigned int WorkAreaRedgeT4;
        unsigned int WorkAreaLedgeT5;
        unsigned int WorkAreaRedgeT5;
        unsigned int WorkAreaLedgeT6;
        unsigned int WorkAreaRedgeT6;
        unsigned int WorkAreaLedgeT7;
        unsigned int WorkAreaRedgeT7;
        unsigned int WorkAreaLedgeT8;
        unsigned int WorkAreaRedgeT8;
        unsigned int WorkAreaLedgeT9;
        unsigned int WorkAreaRedgeT9;
        unsigned int WorkAreaLedgeT0;
        unsigned int WorkAreaRedgeT0;
        unsigned int WorkAreaLedgeTa;
        unsigned int WorkAreaRedgeTa;
        unsigned int WorkAreaLedgeTb;
        unsigned int WorkAreaRedgeTb;
        unsigned int WorkAreaLedgeTc;
        unsigned int WorkAreaRedgeTc;
        unsigned int WorkAreaLedgeTd;
        unsigned int WorkAreaRedgeTd;
        unsigned int WorkAreaLedgeTe;
        unsigned int WorkAreaRedgeTe;
        unsigned int WorkAreaLedgeTf;
        unsigned int WorkAreaRedgeTf;
char llTOaDigits[27]; // 9,223,372,036,854,775,807: 1(sign or carry)+19(digits)+1('\0')+6(,)
char llTOaDigits2[27]; // 9,223,372,036,854,775,807: 1(sign or carry)+19(digits)+1('\0')+6(,)
char llTOaDigits3[27]; // 9,223,372,036,854,775,807: 1(sign or carry)+19(digits)+1('\0')+6(,)

// IP
unsigned char WILDCARD_FAST_flag;
unsigned char WILDCARD_IP_flag;
unsigned char Exact_flag;
unsigned char Dump_flag;
unsigned char EXHAUSTIVE_flag;
int Melnitchka=0;
char *Auberge[4] = {"|\0","/\0","-\0","\\\0"};
int MAXboth;
char *ASCII010 = "\n\0";
int a;
unsigned long FREADclocks=0;
unsigned long long ticksTOTAL=0, ticksStart;

int YesDump;

#ifdef _N_HIGH_PRIORITY
   DWORD dwError, dwPriClass;
#endif

/*
Upper and lower bounds:
The Levenshtein distance has several simple upper and lower bounds that are useful in applications which compute many of them and compare them. These include:
- It is always at least the difference of the sizes of the two strings.
- It is at most the length of the longer string.
- It is zero if and only if the strings are identical.
- If the strings are the same size, the Hamming distance is an upper bound on the Levenshtein distance.
Possible improvements to this algorithm include:
- We can adapt the algorithm to use less space, O(m) instead of O(mn), since it only requires that the previous row and current row be stored at any one time.
- If we are only interested in the distance if it is smaller than a threshold k, then it suffices to compute a diagonal stripe of width 2k+1 in the matrix. In this way, the algorithm can be run in O(kl)
        time, where l is the length of the shortest string.[1]
*/

// A very good resource:
// http://shaunwagner.com/writings_computer_levenshtein.html

        int i,j,m,n,l,BB;
        char s[] = "sitting";
```

```
	char t[] = "kitten";
	m = strlen(s);
	n = strlen(t);
//		k	i	t	t	e	n
//	0	1	2	3	4	5	6
// s	1	1	2	3	4	5	6
// i	2	2	1	2	3	4	5
// t	3	3	2	1	2	3	4
// t	4	4	3	2	1	2	3
// i	5	5	4	3	2	2	3
// n	6	6	5	4	3	3	2
// g	7	7	6	5	4	4	3

/*
	for(i=0;i<=m;i++)
		Levenshtein[i][0] = i;
	for(j=0;j<=n;j++)
		Levenshtein[0][j] = j;
	for (j=1;j<=n;j++) {
		for(i=1;i<=m;i++) {
			if(s[i-1] == t[j-1])
				Levenshtein[i][j] = Levenshtein[i-1][j-1];
			else
				Levenshtein[i][j] = MIN(MIN((Levenshtein[i-1][j]+1),(Levenshtein[i][j-1]+1)),(Levenshtein[i-1][j-1]+1));
		}
	}
	printf("Levenshtein Distance: %d\n", Levenshtein[m][n]);
	exit (0);
*/

/*
 ____  __.                 .__                           
|    |/ _|____  _________  |  |__ _____    ____ _____    
|      <  \__  \ \___   /\__  \ |  |  \\__  \  /    \\__  \   ______
|    |  \  / __ \_/    /  / __ \|   Y  \/ __ \|   |  \/ __ \_ /_____/
|____|__ \(____  /_____ \(____  /___|  (____  /___|  (____  /
        \/     \/      \/     \/     \/     \/     \/     \/ 
___________                     __      /\ __      __.__.__       .___                   .___  /\ ___________                           
\_   _____/__  ________    _____/  |_   / //  \    /  \__|  |    __| _/____ _____ _______  __| _/ / / \_   _____/_ __________________.__.
 |    __)_\  \/  /\__  \ _/ ___\   __\ / / \   \/\/   /  |  |   / __ |/ ___\\__  \\_  __ \/ __ | / /   |    __)|  |  \___   /\___   <   |  |
 |        \>    <  / __ \\  \___|  |  / /   \        /|  |  |__/ /_/ \  \___ / __ \|  | \/ /_/ |/ /    |     \ |  |  //    /  /    / \___  |
/_______  /__/\_ \(____  /\___  >__| / /     \__/\  / |__|____/\____ |\___  >____  /__|  \____ / /     \___  / |____//_____ \/_____ \/ ____|
        \/      \/     \/     \/     \/           \/                \/    \/     \/           \/\/          \/              \/      \/\/     
  _________                           .__                                     
 /   _____/ ____ _____ _______   ____ |  |__   ___________   ____   ______ ______
 \_____  \_/ __ \\__  \\_  __ \_/ ___\|  |  \_/ __ \_  __ \_/ __ \ /  ___//  ___/
 /        \  ___/ / __ \|  | \/\  \___|   Y  \  ___/|  | \/\  ___/ \___ \ \___ \ 
/_______  /\___  >____  /__|    \___  >___|  /\___  >__|    \___  >____  >____  >
        \/     \/     \/            \/     \/     \/            \/     \/     \/ 

*/

printf(" ____  __.                 .__                           \n");
printf("|    |/ _|____  _________  |  |__ _____    ____ _____    \n");
printf("|      <  \\__  \\ \\___   /\\__  \\ |  |  \\\\__  \\  /    \\\\__  \\  \n");
printf("|    |  \\  / __ \\_/    /  / __ \\|   Y  \\/ __ \\|   |  \\/ __ \\_\n");
printf("|____|__ \\(____  /_____ \\(____  /___|  (____  /___|  (____  /\n");
printf("        \\/     \\/      \\/     \\/     \\/     \\/     \\/     \\/ \n");

	printf("Kazahana, a typhoon-class exact & wildcards & Levenshtein Distance (Wagner-Fischer) searcheress, r. 1-+
+fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix_DEFINE_Trolldom_LineByLine_fix_BackToBuffer_Lowercasing-fix, copyleft Kaze 2024-Jan-29.\n");
	if (argc != 4+1 && argc != 3+1) {
	printf("Usage: Kazahana [AtMostLevenshteinDistance][e] string textualfile MasterBufferSize\n");
	printf("Note0: MasterBufferSize is in KB, consider 1024, 3072, 7168 or bigger (up to 2GB). Three additional flags were mapped on this value: all dump\n");
	printf("       lines (except fuzzy's) will have/lack pattern-source info when the number is even/odd respectively, see Examples #5 and #6.\n");
	printf("       When MasterBufferSize ends in 0, then No-Dump i.e. hits are only counted.\n");
	printf("Note0a: Caution! Reported hits are not actual ones but all LINES containing a hit (or hits), e.g. for pattern 'Boom' a line as 'Boom-Boom!' yields one hit not two.\n");
	printf("Note1: There are three regimes: exact, wildcards and fuzzy searches. First two kick in when 3 parameters are given, fuzzy when 4.\n");
	printf("Note2: What decides whether exact or wildcards? Of course presence of at least one wildcard. To see exact search see Example #4.\n");
	printf("Note3: Exact search hits with 'Railgun_Trolldom', not 'Railgun_Sekireigan_Wolfram'.\n");
	printf("Note4a: Incoming string is automatically lowercased for fuzzy searches i.e. they are case insensitive.\n");
	printf("Note4b: Incoming string is NOT automatically lowercased for wildcards searches when MasterBufferSize ends in 0..4 i.e. they are case sensitive.\n");
	printf("Note4c: Incoming string is automatically lowercased for wildcards searches when MasterBufferSize ends in 5..9 i.e. they are case insensitive.\n");
	printf("Note5: Incoming string could be up to %d/%d chars for Exact&Wildcard&ExhaustiveFuzzy/Fuzzy respectively.\n", MaxLineLength +(167*MaxLineLength), MaxLineLength);
	printf("Note5a: Since 2013-Nov-21 Levenshtein search exits not when the incoming line is bigger than %d chars, now it just skips longer lines.\n", MaxLineLength);
	printf("Note5b: Since 2013-Dec-05 Levenshtein search can be EXHAUSTIVE if LD is postfixed with 'e'.\n");
	printf("Note6: Incoming textualfile could be bigger than 4GB.\n");
	printf("Note7: Each line should end with [CR]LF, that is Windows or/and UNIX style.\n");
	printf("Note8: The dump goes to Kazahana.txt file.\n");
	printf("Note9a: Nine SLOW wildcards are available:\n");
	printf("        wildcard '*' any character(s) or empty,\n");
	printf("        wildcard '.' any ALPHA character(s) or empty,\n");
	printf("        wildcard '%%' any NON-ALPHA character(s) or empty,\n");
	printf("        wildcard '@'/'#' any character {or empty}/{and not empty},\n");
	printf("        wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},\n");
	printf("        wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty}.\n");
	printf("Note9b: Two FAST wildcards are available:\n");
	printf("        wildcard '&' any character(s) or empty,\n");
	printf("        wildcard '+' any character and not empty.\n");
```



```
	printf("Note9c: Don't mix SLOW and FAST, the SLOW overrides the FAST, i.e. presence of at least one of the 9 wildcards cancels FAST mode.\n");
	printf("Example1: E:\\>Kazahana 0 ramjet MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536\n");
	printf("Example2: E:\\>Kazahana 3 psychedlicize MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536\n");
	printf("Example3: E:\\>Kazahana %cpsyched^^^^^^ize^%c MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536\n", 34, 34);
	printf("Example4: E:\\>Kazahana %cmetal fatigue%c enwiki-20121201-pages-articles.xml 7168\n", 34, 34);
	printf("Example5: E:\\>Kazahana %cout^^^^^^^^^^^^^ize*%c MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536\n", 34, 34);
	printf("          E:\\>type Kazahana.txt\n");
	printf("          [out^^^^^^^^^^^^^ize*] outhyperbolize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/\n");
	printf("          [out^^^^^^^^^^^^^ize*] outsize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/\n");
	printf("          [out^^^^^^^^^^^^^ize*] outsized /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/\n");
	printf("          [out^^^^^^^^^^^^^ize*] outstrategize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/\n");
	printf("          [out^^^^^^^^^^^^^ize*] outtyrannize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/\n");
	printf("Example6: E:\\>Kazahana %cout^^^^^^^^^^^^^ize*%c MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1537\n", 34, 34);
	printf("          E:\\>type Kazahana.txt\n");
	printf("          outhyperbolize\n");
	printf("          outsize\n");
	printf("          outsized\n");
	printf("          outstrategize\n");
	printf("          outtyrannize\n");
	printf("Example7: E:\\>Kazahana 2e edelvais MASAKARI_General-Purpose_Grade_English_Wordlist.wrd 1024\n");
	printf("          E:\\>type Kazahana.txt\n");
	printf("          bordelais\n");
	printf("          bordelaise\n");
	printf("          edelweiss\n");
	printf("          edelweisses\n");
	printf("          foredevised\n");
	printf("          predellas\n");
	printf("          psychedelicism\n");
	(void) time(&t1);
	(void) time(&t3);
	while (t3 == t1) (void) time(&t3);
	t1=t3;
	clocks1 = clock();
#if defined(_icl_mumbo_jumbo_)
ticksStart = GetRDTSC();
#endif
	while (t3 != t1+2) (void) time(&t3);
#if defined(_icl_mumbo_jumbo_)
ticksTOTAL = ticksTOTAL + GetRDTSC() - ticksStart;
#endif
	printf("Info1: One second seems to have %s clocks.\n", _ui64toaKAZEcomma((clock()-clocks1)/2, llTOaDigits, 10));
#if defined(_icl_mumbo_jumbo_)
	printf("Info2: This CPU seems to be working at %s MHz.\n", _ui64toaKAZEcomma(ticksTOTAL/2/1000000, llTOaDigits, 10));
#endif
	} else {
	if (argc == 3+1) WILDCARD_IP_flag = 1; else WILDCARD_IP_flag = 0;
	(void) time(&t1);
	(void) time(&t3);
	while (t3 == t1) (void) time(&t3);
	t1=t3;
	clocks1 = clock();

if (argc == 4+1) {
	// ASCII 48 is '0'
	// ASCII 52 is '4'
	// ASCII 57 is '9'
	if ( (*(unsigned char *)(argv[4]+strlen(argv[4])-1)-48) > 4) CaseSensitiveWildcardMatching_flag=0; else CaseSensitiveWildcardMatching_flag=1;
	if ( (*(unsigned char *)(argv[4]+strlen(argv[4])-1)-48) == 0) YesDump=0; else YesDump=1;
}
if (argc == 3+1) {
	// ASCII 48 is '0'
	// ASCII 52 is '4'
	// ASCII 57 is '9'
	if ( (*(unsigned char *)(argv[3]+strlen(argv[3])-1)-48) > 4) CaseSensitiveWildcardMatching_flag=0; else CaseSensitiveWildcardMatching_flag=1;
	if ( (*(unsigned char *)(argv[3]+strlen(argv[3])-1)-48) == 0) YesDump=0; else YesDump=1;
}

	//n = strlen(argv[2-WILDCARD_IP_flag]);
	n = 0;
	while (argv[2-WILDCARD_IP_flag][n]) {
	if (CaseSensitiveWildcardMatching_flag == 1)
		wrdARG[ n ] = ( argv[2-WILDCARD_IP_flag][n] );
	else
		wrdARG[ n ] = KAZE_toupper( argv[2-WILDCARD_IP_flag][n] );
		n++;
		if (n>MaxLineLength)
		{ printf( "Kazahana: Incoming xgram exceeding the limit.\n" ); return( 1 ); }
	}
	wrdARG[ n ] = 0; // Needed 'cause wrdARG is not zeroed!

	EXHAUSTIVE_flag = 0;
	if (!WILDCARD_IP_flag) {
		AtMostLevenshteinDistance = atoi(argv[1]);
		if ( KAZE_tolower( argv[1][strlen(argv[1])-1] ) == 'e') EXHAUSTIVE_flag = 1;
	}

	Exact_flag = 0;
	WILDCARD_FAST_flag = 2;
	if (WILDCARD_IP_flag) {
```

```
        if ( memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfreeFAST[0], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfreeFAST[1], n) == NULL ) Exact_flag = 1;
        else WILDCARD_FAST_flag = 1;
        }

        if (EXHAUSTIVE_flag && n<=AtMostLevenshteinDistance)
        { printf( "Kazahana: In EXHAUSTIVE mode 'PatternLength' should be bigger than 'AtMostLevenshteinDistance'.\n" ); return( 1 ); }

        // The way it is done now the SLOW overrides the FAST, meaning that presence of at least one of the 9 wildcards trumps the FAST mode.

        if (WILDCARD_IP_flag) {
        if ( memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[0], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[1], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[2], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[3], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[4], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[5], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[6], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[7], n) == NULL && \
             memchrKAZE(argv[2-WILDCARD_IP_flag], &TAGfree[8], n) == NULL ) Exact_flag = 1;
        else WILDCARD_FAST_flag = 0;
        }

        if (WILDCARD_FAST_flag == 1) Exact_flag = 0;
        if (WILDCARD_FAST_flag == 0) Exact_flag = 0;

        printf("WILDCARD_IP_flag = %u\n", WILDCARD_IP_flag );
        printf("WILDCARD_FAST_flag = %u\n", WILDCARD_FAST_flag );
        printf("Exact_flag = %u\n", Exact_flag );
        printf("EXHAUSTIVE_flag = %u\n", EXHAUSTIVE_flag );

        if (Exact_flag == 1) {
              if (CaseSensitiveWildcardMatching_flag == 1) printf("Enforcing Case-Sensitive Exact mode ...\n");
              if (CaseSensitiveWildcardMatching_flag == 0) printf("Enforcing Case-Insensitive Exact mode ...\n");
        }

        if (WILDCARD_IP_flag == 0) {
              if (EXHAUSTIVE_flag == 1) printf("Enforcing Case-Insensitive Fuzzy (EXHAUSTIVE) mode ...\n");
              if (EXHAUSTIVE_flag == 0) printf("Enforcing Case-Insensitive Fuzzy mode ...\n");
        }

        if (WILDCARD_FAST_flag != 2) {
              if (CaseSensitiveWildcardMatching_flag == 1) printf("Enforcing Case-Sensitive Wildcard mode ...\n");
              if (CaseSensitiveWildcardMatching_flag == 0) printf("Enforcing Case-Insensitive Wildcard mode ...\n");
        }

        if (WILDCARD_FAST_flag == 1) printf("Enforcing FAST Wildcard mode ...\n");
        if (WILDCARD_FAST_flag == 0) printf("Enforcing SLOW Wildcard mode ...\n");

        if (Exact_flag && n==1)
        { printf( "Kazahana: Incoming xgram should be longer than 1 char.\n" ); return( 1 ); }
        if (n>255)
        { printf( "Kazahana: Incoming xgram should be shorter than 256 chars.\n" ); return( 1 ); }

/*
        if (Exact_flag) {
        n = 0;
        while (argv[2-WILDCARD_IP_flag][n]) {
              wrdARG[ n ] = argv[2-WILDCARD_IP_flag][n];
              n++;
              if (n>MaxLineLength)
              { printf( "Kazahana: Incoming xgram exceeding the limit.\n" ); return( 1 ); }
        }
        wrdARG[ n ] = 0; // Needed 'cause wrdARG is not zeroed!
        }
*/

#ifdef _N_HIGH_PRIORITY
   if(!SetPriorityClass(GetCurrentProcess(), HIGH_PRIORITY_CLASS))
   {
//       _tprintf(TEXT("Already REALTIME_PRIORITY.\n"));
//       goto Cleanup;
   }
   if(!SetPriorityClass(GetCurrentProcess(), REALTIME_PRIORITY_CLASS))
   {
//       _tprintf(TEXT("Already REALTIME_PRIORITY.\n"));
//       goto Cleanup;
   }
   // Display priority class

   dwPriClass = GetPriorityClass(GetCurrentProcess());

   //_tprintf(TEXT("Current priority class is 0x%x\n"), dwPriClass);

   if (dwPriClass==0x00000080) printf("Current priority class is HIGH_PRIORITY_CLASS.\n");
   if (dwPriClass==0x00000100) printf("Current priority class is REALTIME_PRIORITY_CLASS.\n");
#endif

printf("Pattern: %s\n",wrdARG);
```

```
// Commented 2019-May-17 [[[[[[[[
/*
        if (Exact_flag) {
// Initializing Gulliver's arrays:
for (a=0; a < 256; a++) {bm_bc[a]=n; bm_bc2nd[a]=n+1;}
for (a=0; a < n-1; a++) bm_bc[argv[2-WILDCARD_IP_flag][a]]=n-a-1;
for (a=0; a < n; a++) bm_bc2nd[argv[2-WILDCARD_IP_flag][a]]=n-a;
for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]= n-1;} // 'memset' if not optimized
for (a=0; a < n-1; a++) bm_Horspool_Order2[*(unsigned short *)(argv[2-WILDCARD_IP_flag]+a)]=a; // Rightmost appearance/position is needed
// Bari arrays:
for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}
//for (a=0; a < n-1; a++) bm_Horspool_Order2[*(unsigned short *)(argv[2-WILDCARD_IP_flag]+a)]=1;
        }
*/
// Commented 2019-May-17 ]]]]]]]]
for (a=0; a < 256*256; a++) {bm_Horspool_Order2[a]=0;}

#ifdef Commence_OpenMP
                printf("omp_get_num_procs( ) = %d\n", omp_get_num_procs( ));
                printf("omp_get_max_threads( ) = %d\n", omp_get_max_threads( ));
                //printf("Enforcing DUAD i.e. double-threads ...\n");
                //printf("Enforcing QUAD i.e. quadruple-threads ...\n");
                //printf("Enforcing OCTAD i.e. octuple-threads ...\n");
                printf("Enforcing HEXADECAD i.e. hexadecuple-threads ...\n");
#else
                printf("Enforcing MONAD i.e. single-thread ...\n");
#endif

/*
Sadly the topic is not fully covered, so here my attempt to fill the gap follows:
noun, adjective, verb, adverb

single/singles/singlet/singlets, single/singled, single/singles/singled/singling, singly
double/doubles/doublet/doublets, double/doubled, double/doubles/doubled/doubling, doubly
triple/triples/triplet/triplets, triple/tripled, triple/triples/tripled/tripling, triply
quadruple/quadruples/quadruplet/quadruplets, quadruple/quadrupled, quadruple/quadruples/quadrupled/quadrupling, quadruply
quintuple/quintuples/quintuplet/quintuplets, quintuple/quintupled, quintuple/quintuples/quintupled/quintupling, quintuply
sextuple/sextuples/sextuplet/sextuplets, sextuple/sextupled, sextuple/sextuples/sextupled/sextupling, sextuply
septuple/septuples/septuplet/septuplets, septuple/septupled, septuple/septuples/septupled/septupling, septuply
octuple/octuples/octuplet/octuplets, octuple/octupled, octuple/octuples/octupled/octupling, octuply
nonuple/nonuples/nonuplet/nonuplets, nonuple/nonupled, nonuple/nonuples/nonupled/nonupling, nonuply
decuple/decuples/decuplet/decuplets, decuple/decupled, decuple/decuples/decupled/decupling, decuply

And one adverbial example given in HERITAGE:
triply
adv.

1. In three ways: As an actor, singer, and juggler, she was triply qualified for the role.
2. To a triple degree: a triply redundant navigational system.
3. Three times: Prices were triply inflated.

It is obvious that other analogs are in use (are preferred):
dual instead of double
quad instead of quadruple
hexa instead of sextuple

This prompts for dumping the *ad analogs (NOUNS!):
monad
duad/dyad
triad
quad/tetrad
pentad
hexad
heptad/hebdomad (for 7)
octad
ennead (for 9)
decad (for 10)
...
duodecad (for 12)
I think my choice hexadecad is even better (it refers to a group) than hexadecuple/sexdecuple couple, don't you think? (for 16)
...
chiliad (for 1000)
myriad (for 10000)
*/

// MAXboth = MaxLineLength +1+1 +(167*WILDCARD_IP_flag*MaxLineLength); // Buggy line, fixed with next one in r. ...CS_fix
        if (WILDCARD_IP_flag) {
                MAXboth = MaxLineLength +1+1 +(167*WILDCARD_IP_flag*MaxLineLength);
        } else {
                MAXboth = MaxLineLength +1+1 +(167*EXHAUSTIVE_flag*MaxLineLength);
        }

if (argc == 4+1) {
        CACHEsize = atoi(argv[4])*(1<<10);
        if (atoi(argv[4]) & 1) Dump_flag=0; else Dump_flag=1;
}
if (argc == 3+1) {
        CACHEsize = atoi(argv[3])*(1<<10);
```

```
      if (atoi(argv[3]) & 1) Dump_flag=0; else Dump_flag=1;
}

memory_size = CACHEsize+65;
printf( "Allocating Master-Buffer %uKB ... ", (memory_size>>10) );
xgamsCACHE = (char *)malloc( memory_size );
if( xgamsCACHE == NULL )
{ puts( "\nKazahana: Needed memory allocation denied!\n" ); return( 1 ); }
printf( "OK\n");
if (64 - (((size_t)xgamsCACHE) % 64) != 0)
      xgamsCACHE = xgamsCACHE + 64 - (((size_t)xgamsCACHE) % 64);

      for(i=0;i<=MaxLineLength;i++) {
            LevenshteinT1[i][0] = i;
            LevenshteinT1[0][i] = i;
            LevenshteinT2[i][0] = i;
            LevenshteinT2[0][i] = i;
            LevenshteinT3[i][0] = i;
            LevenshteinT3[0][i] = i;
            LevenshteinT4[i][0] = i;
            LevenshteinT4[0][i] = i;
            LevenshteinT5[i][0] = i;
            LevenshteinT5[0][i] = i;
            LevenshteinT6[i][0] = i;
            LevenshteinT6[0][i] = i;
            LevenshteinT7[i][0] = i;
            LevenshteinT7[0][i] = i;
            LevenshteinT8[i][0] = i;
            LevenshteinT8[0][i] = i;
            LevenshteinT9[i][0] = i;
            LevenshteinT9[0][i] = i;
            LevenshteinT0[i][0] = i;
            LevenshteinT0[0][i] = i;
            LevenshteinTa[i][0] = i;
            LevenshteinTa[0][i] = i;
            LevenshteinTb[i][0] = i;
            LevenshteinTb[0][i] = i;
            LevenshteinTc[i][0] = i;
            LevenshteinTc[0][i] = i;
            LevenshteinTd[i][0] = i;
            LevenshteinTd[0][i] = i;
            LevenshteinTe[i][0] = i;
            LevenshteinTe[0][i] = i;
            LevenshteinTf[i][0] = i;
            LevenshteinTf[0][i] = i;
      }

// Out of courtesy...
memset (wrdCACHEDT1, 0, MaxLineLength+1+1);
memset (wrdCACHEDT2, 0, MaxLineLength+1+1);
memset (wrdCACHEDT3, 0, MaxLineLength+1+1);
memset (wrdCACHEDT4, 0, MaxLineLength+1+1);
memset (wrdCACHEDT5, 0, MaxLineLength+1+1);
memset (wrdCACHEDT6, 0, MaxLineLength+1+1);
memset (wrdCACHEDT7, 0, MaxLineLength+1+1);
memset (wrdCACHEDT8, 0, MaxLineLength+1+1);
memset (wrdCACHEDT9, 0, MaxLineLength+1+1);
memset (wrdCACHEDT0, 0, MaxLineLength+1+1);
memset (wrdCACHEDTa, 0, MaxLineLength+1+1);
memset (wrdCACHEDTb, 0, MaxLineLength+1+1);
memset (wrdCACHEDTc, 0, MaxLineLength+1+1);
memset (wrdCACHEDTd, 0, MaxLineLength+1+1);
memset (wrdCACHEDTe, 0, MaxLineLength+1+1);
memset (wrdCACHEDTf, 0, MaxLineLength+1+1);

if( ( fp_inLINE = fopen( argv[3-WILDCARD_IP_flag], "rb" ) ) == NULL )
{ printf( "Kazahana: Can't open %s file.\n", argv[3-WILDCARD_IP_flag] ); return( 1 ); }

if( ( fp_outLINE = fopen( "Kazahana.txt", "wb" ) ) == NULL )
{ printf( "Kazahana: Can't open Kazahana.txt file.\n" ); return( 1 ); }

#if defined(_WIN32_ENVIRONMENT_)
   // 64bit:
_lseeki64( fileno(fp_inLINE), 0L, SEEK_END );
size_inLINESIXFOUR = _telli64( fileno(fp_inLINE) );
_lseeki64( fileno(fp_inLINE), 0L, SEEK_SET );
#else
   // 64bit:
fseeko( fp_inLINE, 0L, SEEK_END );
size_inLINESIXFOUR = ftello( fp_inLINE );
fseeko( fp_inLINE, 0L, SEEK_SET );
#endif /* defined(_WIN32_ENVIRONMENT_)  */

// MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING [
size_inLINESIXFOURleftforparsing = size_inLINESIXFOUR;
CACHEremainder = 0;
while (size_inLINESIXFOURleftforparsing >= CACHEsize-CACHEremainder) {
      clocks4 = clock();
#if defined(_icl_mumbo_jumbo_)
```

```
ticksStart = GetRDTSC();
#endif
       fread( xgamsCACHE+CACHEremainder, 1, CACHEsize-CACHEremainder, fp_inLINE );

       // TO-DO: Use the vectorized LOWERcasing that I have [
       if ( CaseSensitiveWildcardMatching_flag==0 ) {
             if ( (CACHEsize-CACHEremainder)%16 ) { //if not padded then scalar
                          for(i=0;i<CACHEsize-CACHEremainder;i++) {
                                       xgamsCACHE[ i ] = KAZE_toupper( xgamsCACHE[ i ] );
                          }
             } else {
                 UCASE_XMM_inplace(xgamsCACHE, CACHEsize-CACHEremainder);
             }
       }
       // TO-DO: Use the vectorized LOWERcasing that I have ]

#if defined(_icl_mumbo_jumbo_)
ticksTOTAL = ticksTOTAL + GetRDTSC() - ticksStart;
#endif
       clocks3 = clock();
       FREADclocks = FREADclocks + (clocks3-clocks4);
       size_inLINESIXFOURleftforparsing = size_inLINESIXFOURleftforparsing - (CACHEsize-CACHEremainder);
       Melnitchka = Melnitchka & 3; // 0 1 2 3: 00 01 10 11
       printf( "%s; Speed: %s bytes/second; Traversed: %s bytes; Dumped: %s\r", Auberge[Melnitchka++], _ui64toaKAZEzerocomma(((size_inLINESIXFOUR-size_inLINESIXFOURleftforparsing)>>0)*CLOCKS_PER_SEC/((long)
       (clocks3 - clocks1 + 1)), llTOaDigits, 10) +12, _ui64toaKAZEcomma((size_inLINESIXFOUR-size_inLINESIXFOURleftforparsing), llTOaDigits2, 10),
       _ui64toaKAZEcomma(DumpedLines1+DumpedLines2+DumpedLines3+DumpedLines4+DumpedLines5+DumpedLines6+DumpedLines7+DumpedLines8+DumpedLines9+DumpedLines0+DumpedLinesa+DumpedLinesb+DumpedLinesc+DumpedLinesd
       +DumpedLinese+DumpedLinesf, llTOaDigits3, 10));
       fflush(stdout);
       CACHEremainder = 0;
       while ( xgamsCACHE[ CACHEsize-1-CACHEremainder ] != 10 ) {
             if ( CACHEsize-1-CACHEremainder == 0 ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
             CACHEremainder++;
       }
       // Working area: xgamsCACHE..xgamsCACHE+CACHEsize-1-CACHEremainder
       //fwrite( xgamsCACHE, 1, xgamsCACHE+CACHEsize-1-CACHEremainder - xgamsCACHE +1, fp_outLINE ); //DELDEL
       //...
// 1st thread 0..1*(MAXxgamsInCACHE/THREADSnumber)-1
// 2nd thread 1*(MAXxgamsInCACHE/THREADSnumber)-1+1..2*(MAXxgamsInCACHE/THREADSnumber)-1
// 3rd thread 2*(MAXxgamsInCACHE/THREADSnumber)-1+1..3*(MAXxgamsInCACHE/THREADSnumber)-1
       // WorkArea pair is the left/right offset in xgamsCACHE pool for each thread, offsets are better than pointers because they are 4bytes not 8
       // CAUTION: An uncrushed bug: the partitions can be without ASCII 010 at all! CRASH!

       //WorkAreaRedgeTe = 15*(CACHEsize/THREADSnumber)-1;
if ( 15*(CACHEsize/THREADSnumber)-1 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX

        WorkAreaLedgeT1 = 0;
       WorkAreaRedgeT1 = 1*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT1 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
       while ( xgamsCACHE[ WorkAreaRedgeT1 ] != 10 ) WorkAreaRedgeT1++; // SLOW!
       /*
       xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT1 ], 10, CACHEsize/THREADSnumber/24*28);
       if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
       // Line below triggers cast warning, no worries since the result of subtraction lies within 'unsigned int' i.e. chunk is always < 4GB:
       WorkAreaRedgeT1 = WorkAreaRedgeT1 + (unsigned int)( (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT1 ]) );
       */
       //printf("%s\n", argv[0] );
       //printf("%s\n", memchr(argv[0], 'K', 300) );
       //printf("%d\n", (unsigned int) (memchr(argv[0], 65, 300)) - (unsigned int)(argv[0]) );
       //Kazahana_r1-+_HEXADECAD-Threads_IntelV12.exe
       //Kazahana_r1-+_HEXADECAD-Threads_IntelV12.exe
       // 17

       WorkAreaLedgeT2 = WorkAreaRedgeT1+1;
       WorkAreaRedgeT2 = 2*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT2 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
       while ( xgamsCACHE[ WorkAreaRedgeT2 ] != 10 ) WorkAreaRedgeT2++;
       /*
       xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT2 ], 10, CACHEsize/THREADSnumber/24*28);
       if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
       WorkAreaRedgeT2 = (unsigned int)( WorkAreaRedgeT2 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT2 ]) );
       */
       WorkAreaLedgeT3 = WorkAreaRedgeT2+1;
       WorkAreaRedgeT3 = 3*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT3 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
       while ( xgamsCACHE[ WorkAreaRedgeT3 ] != 10 ) WorkAreaRedgeT3++;
       /*
       xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT3 ], 10, CACHEsize/THREADSnumber/24*28);
       if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
       WorkAreaRedgeT3 = (unsigned int)( WorkAreaRedgeT3 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT3 ]) );
       */
       WorkAreaLedgeT4 = WorkAreaRedgeT3+1;
       WorkAreaRedgeT4 = 4*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT4 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
       while ( xgamsCACHE[ WorkAreaRedgeT4 ] != 10 ) WorkAreaRedgeT4++;
       /*
       xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT4 ], 10, CACHEsize/THREADSnumber/24*28);
       if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
       WorkAreaRedgeT4 = (unsigned int)( WorkAreaRedgeT4 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT4 ]) );
       */
```

```
        WorkAreaLedgeT5 = WorkAreaRedgeT4+1;
        WorkAreaRedgeT5 = 5*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT5 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeT5 ] != 10 ) WorkAreaRedgeT5++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT5 ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeT5 = (unsigned int)( WorkAreaRedgeT5 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT5 ]) );
        */
        WorkAreaLedgeT6 = WorkAreaRedgeT5+1;
        WorkAreaRedgeT6 = 6*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT6 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeT6 ] != 10 ) WorkAreaRedgeT6++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT6 ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeT6 = (unsigned int)( WorkAreaRedgeT6 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT6 ]) );
        */
        WorkAreaLedgeT7 = WorkAreaRedgeT6+1;
        WorkAreaRedgeT7 = 7*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT7 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeT7 ] != 10 ) WorkAreaRedgeT7++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT7 ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeT7 = (unsigned int)( WorkAreaRedgeT7 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT7 ]) );
        */
        WorkAreaLedgeT8 = WorkAreaRedgeT7+1;
        WorkAreaRedgeT8 = 8*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT8 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeT8 ] != 10 ) WorkAreaRedgeT8++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT8 ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeT8 = (unsigned int)( WorkAreaRedgeT8 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT8 ]) );
        */
        WorkAreaLedgeT9 = WorkAreaRedgeT8+1;
        WorkAreaRedgeT9 = 9*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT9 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeT9 ] != 10 ) WorkAreaRedgeT9++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT9 ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeT9 = (unsigned int)( WorkAreaRedgeT9 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT9 ]) );
        */
        WorkAreaLedgeT0 = WorkAreaRedgeT9+1;
        WorkAreaRedgeT0 = 10*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeT0 >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeT0 ] != 10 ) WorkAreaRedgeT0++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeT0 ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeT0 = (unsigned int)( WorkAreaRedgeT0 + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeT0 ]) );
        */
        WorkAreaLedgeTa = WorkAreaRedgeT0+1;
        WorkAreaRedgeTa = 11*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeTa >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeTa ] != 10 ) WorkAreaRedgeTa++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeTa ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeTa = (unsigned int)( WorkAreaRedgeTa + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeTa ]) );
        */
        WorkAreaLedgeTb = WorkAreaRedgeTa+1;
        WorkAreaRedgeTb = 12*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeTb >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeTb ] != 10 ) WorkAreaRedgeTb++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeTb ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeTb = (unsigned int)( WorkAreaRedgeTb + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeTb ]) );
        */
        WorkAreaLedgeTc = WorkAreaRedgeTb+1;
        WorkAreaRedgeTc = 13*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeTc >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeTc ] != 10 ) WorkAreaRedgeTc++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeTc ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeTc = (unsigned int)( WorkAreaRedgeTc + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeTc ]) );
        */
        WorkAreaLedgeTd = WorkAreaRedgeTc+1;
        WorkAreaRedgeTd = 14*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeTd >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
        while ( xgamsCACHE[ WorkAreaRedgeTd ] != 10 ) WorkAreaRedgeTd++;
        /*
        xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeTd ], 10, CACHEsize/THREADSnumber/24*28);
        if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
        WorkAreaRedgeTd = (unsigned int)( WorkAreaRedgeTd + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeTd ]) );
```

```
       */
       WorkAreaLedgeTe = WorkAreaRedgeTd+1;
       WorkAreaRedgeTe = 15*(CACHEsize/THREADSnumber)-1;
//if ( WorkAreaRedgeTe >= CACHEsize-1-CACHEremainder ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); } // parse BUGfixed: ...nixFIX
       while ( xgamsCACHE[ WorkAreaRedgeTe ] != 10 ) WorkAreaRedgeTe++;
       /*
       xgamsCACHEMEMCHR = memchr(&xgamsCACHE[ WorkAreaRedgeTe ], 10, CACHEsize/THREADSnumber/24*28);
       if ( xgamsCACHEMEMCHR == NULL )  { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
       WorkAreaRedgeTe = (unsigned int)( WorkAreaRedgeTe + (unsigned long long)xgamsCACHEMEMCHR - (unsigned long long)(&xgamsCACHE[ WorkAreaRedgeTe ]) );
       */
       WorkAreaLedgeTf = WorkAreaRedgeTe+1;
        WorkAreaRedgeTf = CACHEsize-1-CACHEremainder;
       if ( WorkAreaLedgeTf >= WorkAreaRedgeTf ) { printf( "\nKazahana: Failure! Too long line encountered. Increase Master-Buffer size.\n" ); return( 1 ); }
       //...

#ifdef Commence_OpenMP
#pragma omp parallel shared(fp_outLINE,n,AtMostLevenshteinDistance,MAXboth,WILDCARD_IP_flag,Exact_flag,WILDCARD_FAST_flag,EXHAUSTIVE_flag,wrdARG)
       private(wrdlen,k,kStartOfLine,StartOfLineFLAG,workbyte,m,wrd,wrdLOW,i,j,l,BB,SkipHeuristic,StartingPosition,FOUNDinPTR)
#endif
{
#ifdef Commence_OpenMP
  #pragma omp sections nowait
#endif
    {

// 1st thread
#ifdef Commence_OpenMP
    #pragma omp section
#endif
       {

//This fragment is bugggy (somehow) [

                                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[[ Since r.1-++
                                                     k = WorkAreaLedgeT1;
                                                     while ( k < WorkAreaRedgeT1 ) {
                                                     #ifdef RG7Gulliver
                                                                  FOUNDinPTR = Railgun_Trolldom_1(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT1-k+1, n); // 2019-May-17
                                                                  //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_1(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT1-k+1, n);
                                                                  //FOUNDinPTR = Railgun_Sekireigan_Wolfram_1(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT1-k+1, n); // 2019-May-17
                                                     #else
                                                                  FOUNDinPTR = Railgun_Quadruplet_7_1(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT1-k+1, n);
                                                     #endif
                                                     // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                     // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                     //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                                     if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                  i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                  while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                  while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                  k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                  //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                  if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                  memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines1++;
                                                                  j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                  j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                     // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP

                                                     #pragma omp critical

#endif
                                                     {
                                                                  if (Dump_flag)
                                                                  fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                  else
                                                                  fprintf( fp_outLINE, "%s\r\n", wrd);
                                                     }

}
                                                                  }
                                                     } else k = WorkAreaRedgeT1;
                                                     } // while
// WHOLE buffer at once not line-by-line ]]]]]]]] Since r.1-++
                                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                        //{

       //memset (wrdCACHEDT1, 0, MaxLineLength+1+1);
       i = 1;
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       wrdlen = 0;
        for( k = WorkAreaLedgeT1; k <= WorkAreaRedgeT1; k++ )
       {
             workbyte = xgamsCACHE[k];

                        if( wrdlen < MAXboth) {
                                                     if (CaseSensitiveWildcardMatching_flag == 0)
```

```
                               wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                      else
                               wrdLOW[ wrdlen ] = ( workbyte );
                                        wrd[ wrdlen ] = workbyte;
                           }
                        if (workbyte == 10) {
                                        TotalLines1++;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                           {
                                        wrd[ wrdlen ] = 0;
                                        wrdLOW[ wrdlen ] = 0;
                                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                      {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                        WordsChecked1++;
                                        if (Exact_flag) {
                                                      //if ((long)( Railgun_Quadruplet_7Gulliver_1(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                      #ifdef RG7Gulliver
                                                                   FOUNDinPTR = Railgun_Trolldom_1(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                                   //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                                                   //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_1(wrdLOW, wrdARG, wrdlen, n);
                                                                   //FOUNDinPTR = Railgun_Sekireigan_Wolfram_1(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                      #else
                                                                   FOUNDinPTR = Railgun_Quadruplet_7_1(wrdLOW, wrdARG, wrdlen, n);
                                                      #endif
                                                      // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                      // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                      //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                                                      if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                                      //if ((long)( Railgun_Quadruplet_7_1(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                                   {
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines1++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                                   }
                                        } else {
                                                      if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                                      if ( WildcardMatch_Iterative_Kaze1(wrdARG, wrdLOW) ) {
#else
                                                      if ( IterativeWildcards1(wrdARG, wrdLOW) ) {
#endif
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines1++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                      }
                                                      } else {
                                                      maskGLOBALlen = n;
                                                      nameGLOBALlen1 = wrdlen;
                                                      if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_1(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines1++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                      }
                                                      }
                                        }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
```

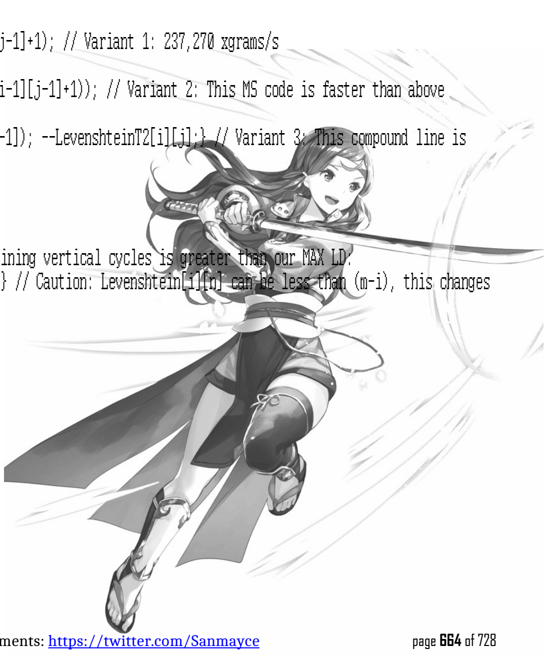
```
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower
//
// Not so trivial, below, when the pattern is one more or one less char the dump fails to match the proper LD:
// Kazahana 2e "Meescha Tate" kazahana3.txt 1536
// Kazahana: Total/Checked/Dumped xgrams: 3/93/0
//
// Kazahana 3e "Meescha Tate" kazahana3.txt 1536
// Kazahana: Total/Checked/Dumped xgrams: 3/34/3
// type Kazahana.txt
// *  1977  &amp;ndash; [[Ameesha Patel]], Indian actress and producer
// * [[Ameesha Patel]]
// * [[Miesha Tate]], Mixed martial artist
//
// Kazahana 2e "Misha Tate" kazahana3.txt 1536
// Kazahana: Total/Checked/Dumped xgrams: 3/74/1
// type Kazahana.txt
// * [[Miesha Tate]], Mixed martial artist
//
// Kazahana 1e "Misha Tate" kazahana3.txt 1536
// Kazahana: Total/Checked/Dumped xgrams: 3/99/0
//
// Kazahana 1e "MieshaA Tate" kazahana3.txt 1536
// Kazahana: Total/Checked/Dumped xgrams: 3/93/0
// Kazahana 2e "MieshaAA Tate" kazahana3.txt 1536
// Kazahana: Total/Checked/Dumped xgrams: 3/90/0

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                        for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                        for (l=0; l < m-BB+1; l++) {
      WordsChecked1++;
// LD [
      SkipHeuristic=0;
      for(i=1;i<=BB;i++) {
            for (j=1;j<=n;j++) {
                        if(wrdLOW[l+i-1] == wrdARG[j-1])
                                    LevenshteinT1[i][j] = LevenshteinT1[i-1][j-1];
                        else
#if defined(_WIN32ASM_)
                                    LevenshteinT1[i][j] = min_AF(LevenshteinT1[i-1][j]+1, LevenshteinT1[i][j-1]+1, LevenshteinT1[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                                    LevenshteinT1[i][j] = MIN(MIN((LevenshteinT1[i-1][j]+1),(LevenshteinT1[i][j-1]+1)),(LevenshteinT1[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
      jumpless code! // 358,327 xgrams/s
                                    //{LevenshteinT1[i][j] = MIN(MIN(LevenshteinT1[i-1][j],LevenshteinT1[i][j-1]),LevenshteinT1[i-1][j-1]); --LevenshteinT1[i][j];} // Variant 3: This compound line is
      much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
            }

            // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
            if ( LevenshteinT1[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT1[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
      nothing the logic is the same.
      }
                                    if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT1[BB][n]) {
                                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines1++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                #pragma omp critical
#endif
                                                {
                                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                                }
}
                                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                                goto EXHAUSTIVE1;
                                                //BB=n+AtMostLevenshteinDistance+1; // Instead of 'goto' setting break for BB 'for', stupidesque
                                                //break; // No need of further checking down the line, one dump only is needed.
                                                //if ((DumpedLines & 0xff) == 0xff)
```

```
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE1:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked1++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT1[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT1[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT1[i][j] = LevenshteinT1[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT1[i][j] = min_AF(LevenshteinT1[i-1][j]+1, LevenshteinT1[i][j-1]+1, LevenshteinT1[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT1[i][j] = MIN(MIN((LevenshteinT1[i-1][j]+1),(LevenshteinT1[i][j-1]+1)),(LevenshteinT1[i-1][j-1]+1)); // Variant 2: This MS code is faster than above jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT1[i][j] = MIN(MIN(LevenshteinT1[i-1][j],LevenshteinT1[i][j-1]),LevenshteinT1[i-1][j-1]); --LevenshteinT1[i][j];} // Variant 3: This compound line is much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT1[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT1[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT1[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines1++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVE1;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDT1, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDT1, &wrdLOW[l], BB); wrdCACHEDT1[BB]=0;
// LD ]
}


                } // if (l < m-BB+1)
                }
                }
EXHAUSTIVE1:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsChecked1++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT1[StartingPosition-1] && wrdCACHEDT1[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
```

```
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinT1[i][j] = LevenshteinT1[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT1[i][j] = min_AF(LevenshteinT1[i-1][j]+1, LevenshteinT1[i][j-1]+1, LevenshteinT1[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT1[i][j] = MIN(MIN((LevenshteinT1[i-1][j]+1),(LevenshteinT1[i][j-1]+1)),(LevenshteinT1[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT1[i][j] = MIN(MIN(LevenshteinT1[i-1][j],LevenshteinT1[i][j-1]),LevenshteinT1[i-1][j-1]); --LevenshteinT1[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT1[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT1[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT1[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines1++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDT1, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
               }
               else wrdlen++;
     } // k 'for'
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        } // if (Exact_flag) {
    }

// 2nd thread
#ifdef Commence_OpenMP
  #pragma omp section
#endif
    {
//This fragment is bugggy (somehow) [

                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                k = WorkAreaLedgeT2;
                                while ( k < WorkAreaRedgeT2 ) {
                                #ifdef RG7Gulliver
                                        FOUNDinPTR = Railgun_Trolldom_2(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT2-k+1, n); // 2019-May-17
                                        //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_2(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT2-k+1, n);
                                        //FOUNDinPTR = Railgun_Sekireigan_Wolfram_2(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT2-k+1, n); // 2019-May-17
                                #else
                                        FOUNDinPTR = Railgun_Quadruplet_7_2(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT2-k+1, n);
                                #endif
                                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                        i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                        while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                        while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                        k = j+1; // Should "point" to first symbol after the dumped fragment.

                                        //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                        if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                        memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines2++;
                                        j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                        j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP
```

```
                                    #pragma omp critical
#endif
                                    {
                                            if (Dump_flag)
                                            fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                            else
                                            fprintf( fp_outLINE, "%s\r\n", wrd);
                                    }
}
                                            }
                                    } else k = WorkAreaRedgeT2;
                                    } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                            } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                            //{

    //memset (wrdCACHEDT2, 0, MaxLineLength+1+1);
    i = 1;
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    wrdlen = 0;
     for( k = WorkAreaLedgeT2; k <= WorkAreaRedgeT2; k++ )
    {
        workbyte = xgamsCACHE[k];

                if( wrdlen < MAXboth) {
                                    if (CaseSensitiveWildcardMatching_flag == 0)
                      wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                    else
                      wrdLOW[ wrdlen ] = ( workbyte );
                            wrd[ wrdlen ] = workbyte;
                  }
                if (workbyte == 10) {
                            TotalLines2++;
// Wildcard search [
                if ( 0 < wrdlen && wrdlen < MAXboth)
                  {
                            wrd[ wrdlen ] = 0;
                            wrdLOW[ wrdlen ] = 0;
                            if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                    {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                            WordsChecked2++;
                            if (Exact_flag) {
                                    //if ((long)( Railgun_Quadruplet_7Gulliver_2(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                    #ifdef RG7Gulliver
                                            FOUNDinPTR = Railgun_Trolldom_2(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                            //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                            //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_2(wrdLOW, wrdARG, wrdlen, n);
                                            //FOUNDinPTR = Railgun_Sekireigan_Wolfram_2(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                    #else
                                            FOUNDinPTR = Railgun_Quadruplet_7_2(wrdLOW, wrdARG, wrdlen, n);
                                    #endif
                                    // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                    // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                    //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                                    if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                    //if ((long)( Railgun_Quadruplet_7_2(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                            {
                                    // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines2++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                    #pragma omp critical
#endif
                                    {
                                            if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                    }
}
                                            }
                            } else {
                                    if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                    if ( WildcardMatch_Iterative_Kaze2(wrdARG, wrdLOW) ) {
#else
                                    if ( IterativeWildcards2(wrdARG, wrdLOW) ) {
#endif
                                    // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines2++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                    #pragma omp critical
#endif
                                    {
                                            if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                    }
```

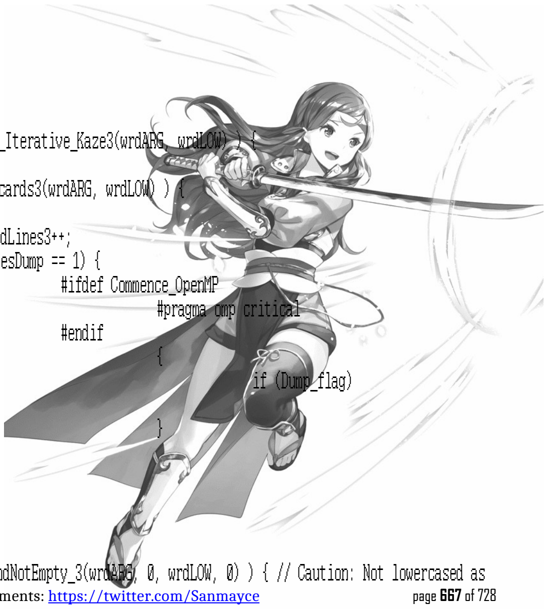
```
}
                }
                } else {
                maskGLOBALlen = n;
                nameGLOBALlen2 = wrdlen;
                if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_2(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines2++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                }
            }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// |Edelweiß || [[edelweiss]] || edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// |Edelweiß |
// Edelweiß ||
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
        for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
        for (l=0; l < m-BB+1; l++) {
  WordsChecked2++;
// LD [
  SkipHeuristic=0;
  for(i=1;i<=BB;i++) {
    for (j=1;j<=n;j++) {
        if(wrdLOW[l+i-1] == wrdARG[j-1])
            LevenshteinT2[i][j] = LevenshteinT2[i-1][j-1];
        else
#if defined(_WIN32ASM_)
            LevenshteinT2[i][j] = min_AF(LevenshteinT2[i-1][j]+1, LevenshteinT2[i][j-1]+1, LevenshteinT2[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
            LevenshteinT2[i][j] = MIN(MIN((LevenshteinT2[i-1][j]+1),(LevenshteinT2[i][j-1]+1)),(LevenshteinT2[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
  jumpless code! // 358,327 xgrams/s
            //{LevenshteinT2[i][j] = MIN(MIN(LevenshteinT2[i-1][j],LevenshteinT2[i][j-1]),LevenshteinT2[i-1][j-1]); --LevenshteinT2[i][j];} // Variant 3: This compound line is
  much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
    }

    // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
    if ( LevenshteinT2[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT2[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
  nothing the logic is the same.
  }
            if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT2[BB][n]) {
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines2++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                goto EXHAUSTIVE2;
```



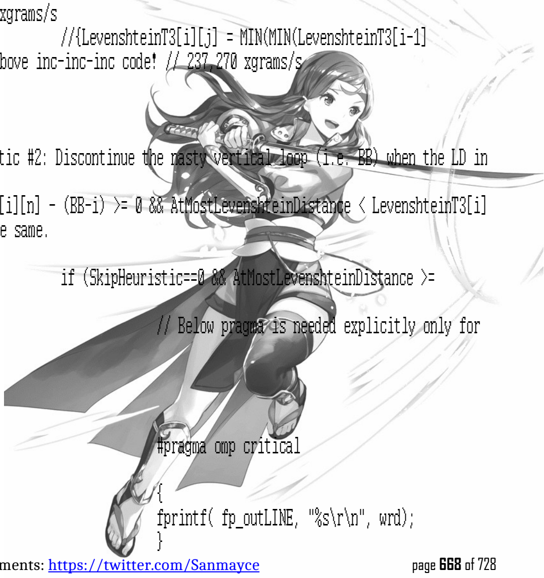
```
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE2:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked2++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT2[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT2[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT2[i][j] = LevenshteinT2[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT2[i][j] = min_AF(LevenshteinT2[i-1][j]+1, LevenshteinT2[i][j-1]+1, LevenshteinT2[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT2[i][j] = MIN(MIN((LevenshteinT2[i-1][j]+1),(LevenshteinT2[i][j-1]+1)),(LevenshteinT2[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT2[i][j] = MIN(MIN(LevenshteinT2[i-1][j],LevenshteinT2[i][j-1]),LevenshteinT2[i-1][j-1]); --LevenshteinT2[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT2[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT2[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT2[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines2++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif

                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVE2;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDT1, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDT2, &wrdLOW[l], BB); wrdCACHEDT2[BB]=0;
// LD ]
}

                } // if (l < m-BB+1)
                }
                }
EXHAUSTIVE2:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsChecked2++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT2[StartingPosition-1] && wrdCACHEDT2[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
```

```
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinT2[i][j] = LevenshteinT2[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT2[i][j] = min_AF(LevenshteinT2[i-1][j]+1, LevenshteinT2[i][j-1]+1, LevenshteinT2[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT2[i][j] = MIN(MIN((LevenshteinT2[i-1][j]+1),(LevenshteinT2[i][j-1]+1)),(LevenshteinT2[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT2[i][j] = MIN(MIN(LevenshteinT2[i-1][j],LevenshteinT2[i][j-1]),LevenshteinT2[i-1][j-1]); --LevenshteinT2[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT2[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT2[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT2[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines2++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDT2, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
               }
               else wrdlen++;
     } // k 'for'
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        } // if (Exact_flag) {
    }


// 3rd thread
#ifdef Commence_OpenMP
  #pragma omp section
#endif
    {
//This fragment is bugggy (somehow) [

        if (Exact_flag) {
                // WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                k = WorkAreaLedgeT3;
                while ( k < WorkAreaRedgeT3 ) {
                        #ifdef RG7Gulliver
                                FOUNDinPTR = Railgun_Trolldom_3(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT3-k+1, n); // 2019-May-17
                                //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_3(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT3-k+1, n);
                                //FOUNDinPTR = Railgun_Sekireigan_Wolfram_3(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT3-k+1, n); // 2019-May-17
                        #else
                                FOUNDinPTR = Railgun_Quadruplet_7_3(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT3-k+1, n);
                        #endif
                        // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                        // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                        //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                        if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                k = j+1; // Should "point" to first symbol after the dumped fragment.
                                //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                        memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines3++;
                                        j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                        j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                        if (YesDump == 1) {
```

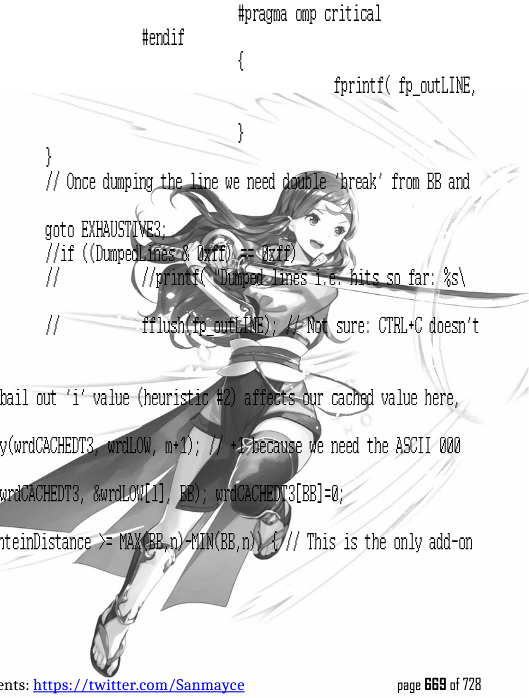
```
                    #ifdef Commence_OpenMP
                        #pragma omp critical
                    #endif
                        {
                            if (Dump_flag)
                            fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                            else
                            fprintf( fp_outLINE, "%s\r\n", wrd);
                        }
                    }
                }
            } else k = WorkAreaRedgeT3;
        } // while
        // WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
    } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
    //{

        //memset (wrdCACHEDT3, 0, MaxLineLength+1+1);
        i = 1;
        wrdlen = 0;
        for( k = WorkAreaLedgeT3; k <= WorkAreaRedgeT3; k++ ) {
            workbyte = xgamsCACHE[k];
      if( wrdlen < MAXboth) {
                if (CaseSensitiveWildcardMatching_flag == 0)
                     wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                else
                     wrdLOW[ wrdlen ] = ( workbyte );
                wrd[ wrdlen ] = workbyte;
            }
      if (workbyte == 10) {
                TotalLines3++;
                // Wildcard search [
                if ( 0 < wrdlen && wrdlen < MAXboth) {
                    wrd[ wrdlen ] = 0;
                    wrdLOW[ wrdlen ] = 0;
                    if ( wrd[ wrdlen-1 ] == 13 ) {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}
                    if (WILDCARD_IP_flag) {
                        // WILDCARD IP [
                        WordsChecked3++;
                        if (Exact_flag) {
                            //if ((long)( Railgun_Quadruplet_7Gulliver_3(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                            #ifdef RG7Gulliver
                                FOUNDinPTR = Railgun_Trolldom_3(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_3(wrdLOW, wrdARG, wrdlen, n);
                                //FOUNDinPTR = Railgun_Sekireigan_Wolfram_3(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                            #else
                                FOUNDinPTR = Railgun_Quadruplet_7_3(wrdLOW, wrdARG, wrdlen, n);
                            #endif
                            // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                            // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                            //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                            if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW)) {
                                //if ((long)( Railgun_Quadruplet_7_3(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                DumpedLines3++;
                                if (YesDump == 1) {
                                    #ifdef Commence_OpenMP
                                        #pragma omp critical
                                    #endif
                                        {
                                            if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n",
  wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
                                }
                            }
                        } else { //if (Exact_flag) {
                            if (WILDCARD_FAST_flag) {
                                #if defined(_WildFastKaze_)
                                    if ( WildcardMatch_Iterative_Kaze3(wrdARG, wrdLOW) ) {
                                #else
                                    if ( IterativeWildcards3(wrdARG, wrdLOW) ) {
                                #endif
                                        DumpedLines3++;
                                        if (YesDump == 1) {
                                            #ifdef Commence_OpenMP
                                                #pragma omp critical
                                            #endif
                                                {
                                                    if (Dump_flag)
  fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                }
                                        }
                                    }
                            } else { //if (WILDCARD_FAST_flag) {
                                maskGLOBALlen = n;
                                nameGLOBALlen3 = wrdlen;
                                if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_3(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as
```



```
it should!
                                DumpedLines3++;
                                if (YesDump == 1) {
                                    #ifdef Commence_OpenMP
                                        #pragma omp critical
                                    #endif
                                        {
                                            if (Dump_flag) fprintf( fp_outLINE, "[%s] %s
/%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
                                }
                            }
                        } //if (WILDCARD_FAST_flag) {
                    } //if (Exact_flag) {
                // WILDCARD IP ]
                } else { //if (WILDCARD_IP_flag) {
                    // A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
                    m = wrdlen; // strlen(wrd);
                    //if (m>MaxLineLength)
                    //{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
                    // Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
                    if (m) {
                    if (EXHAUSTIVE_flag == 1) {

                        // Here we'll walk through the whole length of 'm', ay-yaa.
                        // EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[
                        // Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
                        // We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them
ONE-BY-ONE - a gruelling task indeed!
                        //// Example:
                        // One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
                        // |Edelweiß || [[edelweiss]] || edelweiss flower
                        // Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
                        // To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
                        // 'edelvais' vs 'edelweiss':
                        // - the mispelled one has one character less;
                        // - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
                        // From above we need Building-Blocks of 46 bytes order 8+3.
                        // Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36


                        // The 36 BBs:
                        // |Edelweiß |
                        // Edelweiß ||
                        // ...
                        // weiss flowe
                        // eiss flower

                        // Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                        /*
                                    for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance;
BB++) {
                                    for (l=0; l < m-BB+1; l++) {
                            WordsChecked3++;
                        // LD [
                            SkipHeuristic=0;
                            for(i=1;i<=BB;i++) {
                                for (j=1;j<=n;j++) {
                                    if(wrdLOW[l+i-1] == wrdARG[j-1])
                                        LevenshteinT3[i][j] = LevenshteinT3[i-1][j-1];
                                    else
                        #if defined(_WIN32ASM_)
                                        LevenshteinT3[i][j] = min_AF(LevenshteinT3[i-1][j]+1,
LevenshteinT3[i][j-1]+1, LevenshteinT3[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
                        #else
                                        LevenshteinT3[i][j] = MIN(MIN((LevenshteinT3[i-1][j]+1),
(LevenshteinT3[i][j-1]+1)),(LevenshteinT3[i-1][j-1]+1)); // Variant 2: This MS code is faster than above jumpless code! // 358,327 xgrams/s
                                        //{LevenshteinT3[i][j] = MIN(MIN(LevenshteinT3[i-1]
[j],LevenshteinT3[i][j-1]),LevenshteinT3[i-1][j-1]); --LevenshteinT3[i][j];} // Variant 3: This compound line is much slower than above inc-inc-inc code! // 237,270 xgrams/s
                        #endif
                                }

                                // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in
cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
                                if ( LevenshteinT3[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT3[i]
[n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes nothing the logic is the same.
                            }
                                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >=
LevenshteinT3[BB][n]) {
                                            // Below pragma is needed explicitly only for
MinGW, grrr...
                        DumpedLines3++;
                        if (YesDump == 1) {
                        #ifdef Commence_OpenMP
                                            #pragma omp critical
                        #endif
                                            {
                                            fprintf( fp_outLINE, "%s\r\n", wrd);
                                            }
```



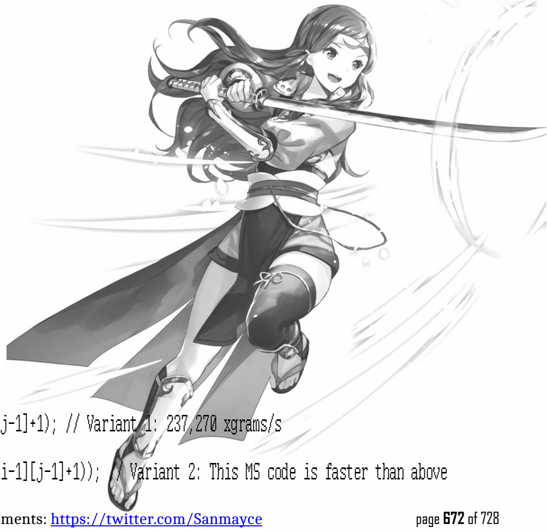
```
                    }
                                        goto EXHAUSTIVE3;
                                        //break; // No need of further checking down
the line, one dump only is needed.
                                        //if ((DumpedLines & 0xff) == 0xff)
                                        //          //printf( "Dumped lines i.e. hits
so far: %s\r", _ui64toaKAZEcomma(DumpedLines, l1TOaDigits, 10) );
                                        //          fflush(fp_outLINE); // Not sure:
CTRL+C doesn't flush?!
                                    }
                    // LD ]
                                }
                                }
                    EXHAUSTIVE3:;
                    */
                    // Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

                    // Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                    for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                    for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                        // From here on 'm' will be replaced by BB
                        if (l < m-BB+1) {
                            if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) { // This is the only add-on for
r.1+
                                WordsChecked3++;
                                // LD [
                                StartingPosition = 1;
                                while (     wrdCACHEDT3[StartingPosition-1] &&
wrdLOW[l+StartingPosition-1] && wrdCACHEDT3[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
                                    StartingPosition++;
                                // The bail out 'i' value (heuristic #2) affects our cached value here,
'StartingPosition' cannot be greater than 'i':
                                StartingPosition = MIN(StartingPosition, i);
                                SkipHeuristic=0;
                                for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
                                    for (j=1;j<=n;j++) {
                                        if(wrdLOW[l+i-1] == wrdARG[j-1])
                                            LevenshteinT3[i][j] =
LevenshteinT3[i-1][j-1];
                                        else
                                            #if defined(_WIN32ASM_)
                                                LevenshteinT3[i][j]
= min_AF(LevenshteinT3[i-1][j]+1, LevenshteinT3[i][j-1]+1, LevenshteinT3[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
                                            #else
                                                LevenshteinT3[i][j]
= MIN(MIN((LevenshteinT3[i-1][j]+1),(LevenshteinT3[i][j-1]+1)),(LevenshteinT3[i-1][j-1]+1)); // Variant 2: This MS code is faster than above jumpless code! // 358,327 xgrams/s
                                                //{LevenshteinT3[i]
[j] = MIN(MIN(LevenshteinT3[i-1][j],LevenshteinT3[i][j-1]),LevenshteinT3[i-1][j-1]); --LevenshteinT3[i][j];} // Variant 3: This compound line is much slower than above inc-inc-inc code! // 237,270
xgrams/s
                                            #endif
                                    }
                                    // A simple heuristic #2: Discontinue the nasty vertical
loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
                                    if ( LevenshteinT3[i][n] - (BB-i) >= 0 &&
AtMostLevenshteinDistance < LevenshteinT3[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes nothing the logic is the same.
                                }
                                if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT3[BB]
[n]) {
                                    DumpedLines3++;
                                    if (YesDump == 1) {
                                        #ifdef Commence_OpenMP
                                            #pragma omp critical
                                        #endif
                                            {
                                                fprintf( fp_outLINE,
"%s\r\n", wrd);
                                            }
                                    }
                                    // Once dumping the line we need double 'break' from BB and
l 'for's:
                                    goto EXHAUSTIVE3;
                                    //if ((DumpedLines & 0xff) == 0xff)
                                    //      //printf( "Dumped lines i.e. hits so far: %s\
r", _ui64toaKAZEcomma(DumpedLines, l1TOaDigits, 10) );
                                    //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't
flush?!
                                }
                                // The bail out 'i' value (heuristic #2) affects our cached value here,
'i' is the needed one:
                                //memcpy(wrdCACHEDT3, wrdLOW, m+1); // +1 because we need the ASCII 000
termination;
                                memcpy(wrdCACHEDT3, &wrdLOW[l], BB); wrdCACHEDT3[BB]=0;
                                // LD ]
                            } //if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) { // This is the only add-on
for r.1+
                        } // if (l < m-BB+1)
                    }
                    }
                    EXHAUSTIVE3:;
```



```
                            // Fast! ])))))))))))))))))))))))))))))))))))))))

                            // EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

                        } else { //if (EXHAUSTIVE_flag == 1) {
                            if (m<=MaxLineLength) {
                            if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) { // This is the only add-on for r.1+
                                WordsChecked3++;
                                // LD [
                                StartingPosition = 1;
                                while (      wrdCACHEDT3[StartingPosition-1] && wrdCACHEDT3[StartingPosition-
  1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
                                    StartingPosition++;
                                // The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be
  greater than 'i':
                                StartingPosition = MIN(StartingPosition, i);
                                SkipHeuristic=0;
                                for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
                                    for (j=1;j<=n;j++) {
                                        if(wrdLOW[i-1] == wrdARG[j-1])
                                            LevenshteinT3[i][j] = LevenshteinT3[i-1][j-1];
                                        else
                                            #if defined(_WIN32ASM_)
                                                LevenshteinT3[i][j] = min_AF(LevenshteinT3[i-1]
  [j]+1, LevenshteinT3[i][j-1]+1, LevenshteinT3[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
                                            #else
                                                LevenshteinT3[i][j] = MIN(MIN((LevenshteinT3[i-
  1][j]+1),(LevenshteinT3[i][j-1]+1)),(LevenshteinT3[i-1][j-1]+1)); // Variant 2: This MS code is faster than above jumpless code! // 358,327 xgrams/s
                                                //{LevenshteinT3[i][j] =
  MIN(MIN(LevenshteinT3[i-1][j],LevenshteinT3[i][j-1]),LevenshteinT3[i-1][j-1]); --LevenshteinT3[i][j];} // Variant 3: This compound line is much slower than above inc-inc-inc code! // 237,270 xgrams/s
                                            #endif
                                    }
                                    // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in
  cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
                                    if ( LevenshteinT3[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT3[i]
  [n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes nothing the logic is the same.
                                }
                                if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT3[m][n]) {
                                    DumpedLines3++;
                                    if (YesDump == 1) {
                                        #ifdef Commence_OpenMP
                                            #pragma omp critical
                                        #endif
                                            {
                                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                            }
                                    }
                                    //if ((DumpedLines & 0xff) == 0xff)
                                    //          //printf( "Dumped lines i.e. hits so far: %s\r",
  _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                    //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                }
                                // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
                                memcpy(wrdCACHEDT3, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
                                // LD ]
                            } //if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) { // This is the only add-on for r.1+
                            }
                        } //if (EXHAUSTIVE_flag == 1)
                        }
                    } //if (WILDCARD_IP_flag)
                } //if ( 0 < wrdlen && wrdlen < MAXboth) {
                // Wildcard search ]
                wrdlen = 0;
            } //if (workbyte == 10) {
            else wrdlen++;
    } // k 'for'
  } // if (Exact_flag) {
  } //#pragma



// 4th thread
#ifdef Commence_OpenMP
 #pragma omp section
#endif
  {
//This fragment is bugggy (somehow) [

            if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[[ Since r.1-++
                k = WorkAreaLedgeT4;
                while ( k < WorkAreaRedgeT4 ) {
                #ifdef RG7Gulliver
                    FOUNDinPTR = Railgun_Trolldom_4(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT4-k+1, n); // 2019-May-17
                    //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_4(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT4-k+1, n);
                    //FOUNDinPTR = Railgun_Sekireigan_Wolfram_4(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT4-k+1, n); // 2019-May-17
                #else
                    FOUNDinPTR = Railgun_Quadruplet_7_4(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT4-k+1, n);
```

```
                            #endif
                            // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                            // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                            //if ((long)( FOUNDinPTR - &xgamsCACHE[k] ))=0) {
                            if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                    i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                    while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                    while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                    k = j+1; // Should "point" to first symbol after the dumped fragment.

                                    //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                    if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                    memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines4++;
                                    j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                    j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                            // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP
                            #pragma omp critical
#endif
                            {
                                    if (Dump_flag)
                                    fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                    else
                                    fprintf( fp_outLINE, "%s\r\n", wrd);
                            }
}

                                    }
                            } else k = WorkAreaRedgeT4;
                            } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                    } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                    //{

    //memset (wrdCACHEDT4, 0, MaxLineLength+1+1);
    i = 1;
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    wrdlen = 0;
     for( k = WorkAreaLedgeT4; k <= WorkAreaRedgeT4; k++ )
    {
          workbyte = xgamsCACHE[k];

                if( wrdlen < MAXboth) {
                            if (CaseSensitiveWildcardMatching_flag == 0)
                  wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                            else
                  wrdLOW[ wrdlen ] = ( workbyte );
                      wrd[ wrdlen ] = workbyte;
                 }
                if (workbyte == 10) {
                      TotalLines4++;
// Wildcard search [
                if ( 0 < wrdlen && wrdlen < MAXboth)
                 {
                      wrd[ wrdlen ] = 0;
                      wrdLOW[ wrdlen ] = 0;
                      if ( wrd[ wrdlen-1 ] == 13 ) //CR
                            {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                      WordsChecked4++;
                      if (Exact_flag) {
                            //if ((long)( Railgun_Quadruplet_7Gulliver_4(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))=0)
                            #ifdef RG7Gulliver
                                    FOUNDinPTR = Railgun_Trolldom_4(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                    //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                    //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_4(wrdLOW, wrdARG, wrdlen, n);
                                    //FOUNDinPTR = Railgun_Sekireigan_Wolfram_4(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                            #else
                                    FOUNDinPTR = Railgun_Quadruplet_7_4(wrdLOW, wrdARG, wrdlen, n);
                            #endif
                            // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                            // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                            //if ((long)( FOUNDinPTR - wrdLOW ))=0)
                            if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                            //if ((long)( Railgun_Quadruplet_7_4(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))=0)
                                    {
                            // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines4++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                            #pragma omp critical
#endif
                            {
                                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                            }
```

```
}
                                                                   }
                                        } else {
                                                      if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                                      if ( WildcardMatch_Iterative_Kaze4(wrdARG, wrdLOW) ) {
#else
                                                      if ( IterativeWildcards4(wrdARG, wrdLOW) ) {
#endif
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines4++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                      }
                                                      } else {
                                                      maskGLOBALlen = n;
                                                      nameGLOBALlen4 = wrdlen;
                                                      if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_4(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                                                                                                         // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines4++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                      }
                                                      }
                                        }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':



// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                           for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                           for (l=0; l < m-BB+1; l++) {
       WordsChecked4++;
// LD [
       SkipHeuristic=0;
       for(i=1;i<=BB;i++) {
             for (j=1;j<=n;j++) {
                           if(wrdLOW[l+i-1] == wrdARG[j-1])
                                        LevenshteinT4[i][j] = LevenshteinT4[i-1][j-1];
                           else
#if defined(_WIN32ASM_)
                                        LevenshteinT4[i][j] = min_AF(LevenshteinT4[i-1][j]+1, LevenshteinT4[i][j-1]+1, LevenshteinT4[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                                        LevenshteinT4[i][j] = MIN(MIN((LevenshteinT4[i-1][j]+1),(LevenshteinT4[i][j-1]+1)),(LevenshteinT4[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
       jumpless code! // 358,327 xgrams/s
```



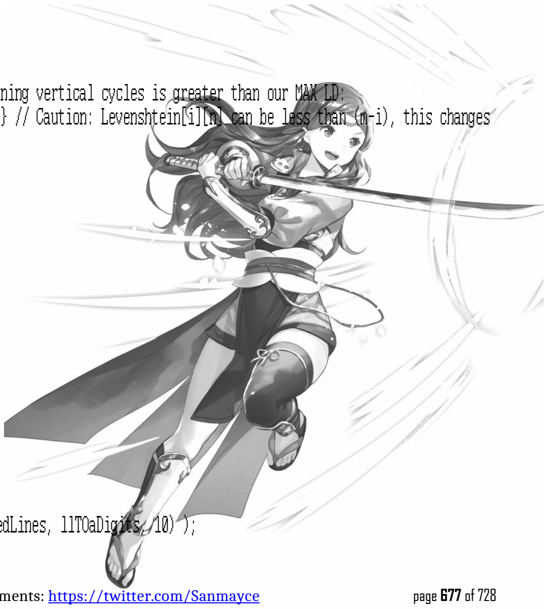
```
                        //{LevenshteinT4[i][j] = MIN(MIN(LevenshteinT4[i-1][j],LevenshteinT4[i][j-1]),LevenshteinT4[i-1][j-1]); --LevenshteinT4[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT4[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT4[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT4[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines4++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVE4;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE4:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked4++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT4[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT4[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT4[i][j] = LevenshteinT4[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT4[i][j] = min_AF(LevenshteinT4[i-1][j]+1, LevenshteinT4[i][j-1]+1, LevenshteinT4[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT4[i][j] = MIN(MIN((LevenshteinT4[i-1][j]+1),(LevenshteinT4[i][j-1]+1)),(LevenshteinT4[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT4[i][j] = MIN(MIN(LevenshteinT4[i-1][j],LevenshteinT4[i][j-1]),LevenshteinT4[i-1][j-1]); --LevenshteinT4[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT4[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT4[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT4[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines4++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVE4;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDT4, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
```

```
      memcpy(wrdCACHEDT4, &wrdLOW[1], BB); wrdCACHEDT4[BB]=0;
// LD ]
}

                      } // if (1 < m-BB+1)
                      }
                      }
EXHAUSTIVE4:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
      WordsChecked4++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT4[StartingPosition-1] && wrdCACHEDT4[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
      StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

      SkipHeuristic=0;
      for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
           for (j=1;j<=n;j++) {
                      if(wrdLOW[i-1] == wrdARG[j-1])
                                 LevenshteinT4[i][j] = LevenshteinT4[i-1][j-1];
                      else
#if defined(_WIN32ASM_)
                                 LevenshteinT4[i][j] = min_AF(LevenshteinT4[i-1][j]+1, LevenshteinT4[i][j-1]+1, LevenshteinT4[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                                 LevenshteinT4[i][j] = MIN(MIN((LevenshteinT4[i-1][j]+1),(LevenshteinT4[i][j-1]+1)),(LevenshteinT4[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
      jumpless code! // 358,327 xgrams/s
                                 //{LevenshteinT4[i][j] = MIN(MIN(LevenshteinT4[i-1][j],LevenshteinT4[i][j-1]),LevenshteinT4[i-1][j-1]); --LevenshteinT4[i][j];} // Variant 3: This compound line is
      much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
           }

           // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
           if ( LevenshteinT4[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT4[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
      nothing the logic is the same.
      }
                                 if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT4[m][n]) {
                                           // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines4++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                           #pragma omp critical
#endif
                                           {
                                           fprintf( fp_outLINE, "%s\r\n", wrd);
                                           }
}
                                           //if ((DumpedLines & 0xff) == 0xff)
                                           //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                           //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                 }
      // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
      memcpy(wrdCACHEDT4, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                      }
// Wildcard search ]
                                 wrdlen = 0;
                    }
                    else wrdlen++;
      } // k 'for'
      //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                 } // if (Exact_flag) {
      }

// 5th thread
#ifdef Commence_OpenMP
   #pragma omp section
#endif
      {
//This fragment is buggy (somehow) [

                                 if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[[ Since r.1-++
                                           k = WorkAreaLedgeT5;
                                           while ( k < WorkAreaRedgeT5 ) {
                                           #ifdef RG7Gulliver
```

```
                                                                    FOUNDinPTR = Railgun_Trolldom_5(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT5-k+1, n); // 2019-May-17
                                                                    //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_5(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT5-k+1, n);
                                                                    //FOUNDinPTR = Railgun_Sekireigan_Wolfram_5(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT5-k+1, n); // 2019-May-17
                                                      #else
                                                                    FOUNDinPTR = Railgun_Quadruplet_7_5(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT5-k+1, n);
                                                      #endif
                                                      // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                      // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                      //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                                      if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                    i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                    while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                    while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                    k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                    //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                    if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                    memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines5++;
                                                                    j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                    j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP

                                                      #pragma omp critical
#endif
                                                      {
                                                                    if (Dump_flag)
                                                                    fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                    else
                                                                    fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}

                                                                    }
                                                      } else k = WorkAreaRedgeT5;
                                                      } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                                         } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                         //{

       //memset (wrdCACHEDT5, 0, MaxLineLength+1+1);
       i = 1;
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       wrdlen = 0;
        for( k = WorkAreaLedgeT5; k <= WorkAreaRedgeT5; k++ )
       {
             workbyte = xgamsCACHE[k];

                        if( wrdlen < MAXboth) {
                                                      if (CaseSensitiveWildcardMatching_flag == 0)
                                wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                      else
                                wrdLOW[ wrdlen ] = ( workbyte );
                                         wrd[ wrdlen ] = workbyte;
                           }
                        if (workbyte == 10) {
                                         TotalLines5++;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                           {
                                         wrd[ wrdlen ] = 0;
                                         wrdLOW[ wrdlen ] = 0;
                                         if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                      {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                         WordsChecked5++;
                                         if (Exact_flag) {
                                                      //if ((long)( Railgun_Quadruplet_7Gulliver_5(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                      #ifdef RG7Gulliver
                                                                    FOUNDinPTR = Railgun_Trolldom_5(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                                    //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                                                    //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_5(wrdLOW, wrdARG, wrdlen, n);
                                                                    //FOUNDinPTR = Railgun_Sekireigan_Wolfram_5(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                      #else
                                                                    FOUNDinPTR = Railgun_Quadruplet_7_5(wrdLOW, wrdARG, wrdlen, n);
                                                      #endif
                                                      // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                      // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                      //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                                                      if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                                      //if ((long)( Railgun_Quadruplet_7_5(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                                    {
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines5++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
```

```
                                        #pragma omp critical
#endif
                                        {
                                                if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                                }
                                } else {
                                        if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                        if ( WildcardMatch_Iterative_Kaze5(wrdARG, wrdLOW) ) {
#else
                                        if ( IterativeWildcards5(wrdARG, wrdLOW) ) {
#endif
                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines5++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                        }
                                        } else {
                                        maskGLOBALlen = n;
                                        nameGLOBALlen5 = wrdlen;
                                        if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_5(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines5++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                        }
                                        }
                                }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                for (l=0; l < m-BB+1; l++) {
    WordsChecked5++;
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT5[i][j] = LevenshteinT5[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT5[i][j] = min_AF(LevenshteinT5[i-1][j]+1, LevenshteinT5[i][j-1]+1, LevenshteinT5[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
```

```
                        LevenshteinT5[i][j] = MIN(MIN((LevenshteinT5[i-1][j]+1),(LevenshteinT5[i][j-1]+1)),(LevenshteinT5[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT5[i][j] = MIN(MIN(LevenshteinT5[i-1][j],LevenshteinT5[i][j-1]),LevenshteinT5[i-1][j-1]); --LevenshteinT5[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT5[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT5[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT5[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines5++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVE5;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE5:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked5++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT5[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT5[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT5[i][j] = LevenshteinT5[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT5[i][j] = min_AF(LevenshteinT5[i-1][j]+1, LevenshteinT5[i][j-1]+1, LevenshteinT5[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT5[i][j] = MIN(MIN((LevenshteinT5[i-1][j]+1),(LevenshteinT5[i][j-1]+1)),(LevenshteinT5[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT5[i][j] = MIN(MIN(LevenshteinT5[i-1][j],LevenshteinT5[i][j-1]),LevenshteinT5[i-1][j-1]); --LevenshteinT5[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT5[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT5[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT5[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines5++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVE5;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
```



```
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDT5, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDT5, &wrdLOW[1], BB); wrdCACHEDT5[BB]=0;
// LD ]
}

                } // if (1 < m-BB+1)
                }
                }
EXHAUSTIVE5:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsChecked5++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT5[StartingPosition-1] && wrdCACHEDT5[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinT5[i][j] = LevenshteinT5[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT5[i][j] = min_AF(LevenshteinT5[i-1][j]+1, LevenshteinT5[i][j-1]+1, LevenshteinT5[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT5[i][j] = MIN(MIN((LevenshteinT5[i-1][j]+1),(LevenshteinT5[i][j-1]+1)),(LevenshteinT5[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT5[i][j] = MIN(MIN(LevenshteinT5[i-1][j],LevenshteinT5[i][j-1]),LevenshteinT5[i-1][j-1]); --LevenshteinT5[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT5[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT5[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT5[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines5++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDT5, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
               }
               else wrdlen++;
     } // k 'for'
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        } // if (Exact_flag) {
    }

// 6th thread
#ifdef Commence_OpenMP
   #pragma omp section
#endif
    {
//This fragment is bugggy (somehow) [

                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                k = WorkAreaLedgeT6;
```

```
                                while ( k < WorkAreaRedgeT6 ) {
                                #ifdef RG7Gulliver
                                        FOUNDinPTR = Railgun_Trolldom_6(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT6-k+1, n); // 2019-May-17
                                        //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_6(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT6-k+1, n);
                                        //FOUNDinPTR = Railgun_Sekireigan_Wolfram_6(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT6-k+1, n); // 2019-May-17
                                #else
                                        FOUNDinPTR = Railgun_Quadruplet_7_6(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT6-k+1, n);
                                #endif
                                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                //if ((long)( FOUNDinPTR - &xgamsCACHE[k] ))=0) {
                                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                        i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                        while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                        while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                        k = j+1; // Should "point" to first symbol after the dumped fragment.

                                        //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                        if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                        memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines6++;
                                        j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                        j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP

                                #pragma omp critical
#endif
                                {
                                        if (Dump_flag)
                                        fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                        else
                                        fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}

                                        }
                                } else k = WorkAreaRedgeT6;
                                } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                        //{

    //memset (wrdCACHEDT6, 0, MaxLineLength+1+1);
    i = 1;
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    wrdlen = 0;
     for( k = WorkAreaLedgeT6; k <= WorkAreaRedgeT6; k++ )
    {
        workbyte = xgamsCACHE[k];

                if( wrdlen < MAXboth) {
                                if (CaseSensitiveWildcardMatching_flag == 0)
                    wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                else
                    wrdLOW[ wrdlen ] = ( workbyte );
                        wrd[ wrdlen ] = workbyte;
                  }
                if (workbyte == 10) {
                        TotalLines6++;
// Wildcard search [
                if ( 0 < wrdlen && wrdlen < MAXboth)
                  {
                        wrd[ wrdlen ] = 0;
                        wrdLOW[ wrdlen ] = 0;
                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                        WordsChecked6++;
                        if (Exact_flag) {
                                //if ((long)( Railgun_Quadruplet_7Gulliver_6(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))=0)
                                #ifdef RG7Gulliver
                                        FOUNDinPTR = Railgun_Trolldom_6(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                        //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                        //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_6(wrdLOW, wrdARG, wrdlen, n);
                                        //FOUNDinPTR = Railgun_Sekireigan_Wolfram_6(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                #else
                                        FOUNDinPTR = Railgun_Quadruplet_7_6(wrdLOW, wrdARG, wrdlen, n);
                                #endif
                                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                //if ((long)( FOUNDinPTR - wrdLOW ))=0)
                                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                //if ((long)( Railgun_Quadruplet_7_6(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))=0)
                                        {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines6++;
```

```
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                                   }
                                        } else {
                                                      if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                                      if ( WildcardMatch_Iterative_Kaze6(wrdARG, wrdLOW) ) {
#else
                                                      if ( IterativeWildcards6(wrdARG, wrdLOW) ) {
#endif
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines6++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                      }
                                                      } else {
                                                      maskGLOBALlen = n;
                                                      nameGLOBALlen6 = wrdlen;
                                                      if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_6(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines6++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                      }
                                                      }
                                        }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                           for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                           for (l=0; l < m-BB+1; l++) {
       WordsChecked6++;
// LD [
       SkipHeuristic=0;
       for(i=1;i<=BB;i++) {
             for (j=1;j<=n;j++) {
                           if(wrdLOW[l+i-1] == wrdARG[j-1])
                                        LevenshteinT6[i][j] = LevenshteinT6[i-1][j-1];
                           else
#if defined(_WIN32ASM_)
```

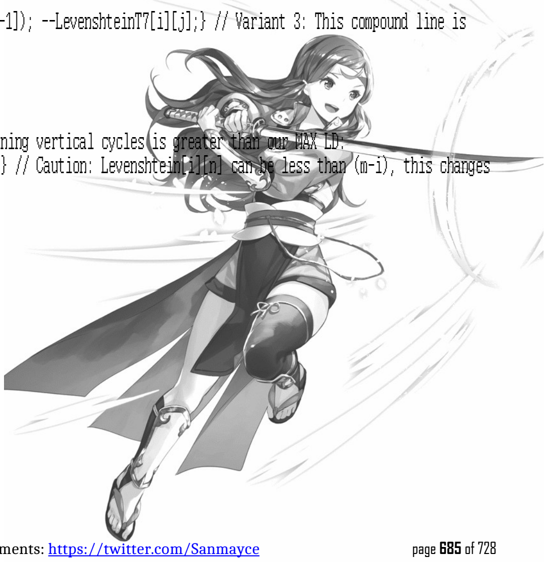
```
                        LevenshteinT6[i][j] = min_AF(LevenshteinT6[i-1][j]+1, LevenshteinT6[i][j-1]+1, LevenshteinT6[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT6[i][j] = MIN(MIN((LevenshteinT6[i-1][j]+1),(LevenshteinT6[i][j-1]+1)),(LevenshteinT6[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT6[i][j] = MIN(MIN(LevenshteinT6[i-1][j],LevenshteinT6[i][j-1]),LevenshteinT6[i-1][j-1]); --LevenshteinT6[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT6[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT6[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT6[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines6++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVE6;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE6:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked6++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT6[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT6[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT6[i][j] = LevenshteinT6[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT6[i][j] = min_AF(LevenshteinT6[i-1][j]+1, LevenshteinT6[i][j-1]+1, LevenshteinT6[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT6[i][j] = MIN(MIN((LevenshteinT6[i-1][j]+1),(LevenshteinT6[i][j-1]+1)),(LevenshteinT6[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT6[i][j] = MIN(MIN(LevenshteinT6[i-1][j],LevenshteinT6[i][j-1]),LevenshteinT6[i-1][j-1]); --LevenshteinT6[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT6[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT6[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT6[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines6++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVE6;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
```

```
                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDT6, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDT6, &wrdLOW[1], BB); wrdCACHEDT6[BB]=0;
// LD ]
}

                } // if (1 < m-BB+1)
                }
                }
EXHAUSTIVE6:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsChecked6++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT6[StartingPosition-1] && wrdCACHEDT6[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinT6[i][j] = LevenshteinT6[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT6[i][j] = min_AF(LevenshteinT6[i-1][j]+1, LevenshteinT6[i][j-1]+1, LevenshteinT6[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT6[i][j] = MIN(MIN((LevenshteinT6[i-1][j]+1),(LevenshteinT6[i][j-1]+1)),(LevenshteinT6[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT6[i][j] = MIN(MIN(LevenshteinT6[i-1][j],LevenshteinT6[i][j-1]),LevenshteinT6[i-1][j-1]); --LevenshteinT6[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT6[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT6[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT6[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines6++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDT6, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
              }
              else wrdlen++;
     } // k 'for'
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        } // if (Exact_flag) {
    }

// 7th thread
#ifdef Commence_OpenMP
   #pragma omp section
#endif
    {
//This fragment is buggy (somehow) [

                        if (Exact_flag) {
```

```
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                        k = WorkAreaLedgeT7;
                                        while ( k < WorkAreaRedgeT7 ) {
                                        #ifdef RG7Gulliver
                                                  FOUNDinPTR = Railgun_Trolldom_7(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT7-k+1, n); // 2019-May-17
                                                  //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_7(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT7-k+1, n);
                                                  //FOUNDinPTR = Railgun_Sekireigan_Wolfram_7(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT7-k+1, n); // 2019-May-17
                                        #else
                                                  FOUNDinPTR = Railgun_Quadruplet_7_7(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT7-k+1, n);
                                        #endif
                                        // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                        // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                        //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                        if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                  i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                  while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                  while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                  k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                  //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                  if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                  memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines7++;
                                                  j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                  j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                        // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                  if (Dump_flag)
                                                  fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                  else
                                                  fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                                  }
                                        } else k = WorkAreaRedgeT7;
                                        } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                              } else { // if (Exact_flag) {

//This fragment is buggy (somehow) ]
                              //{

     //memset (wrdCACHEDT7, 0, MaxLineLength+1+1);
     i = 1;
      //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     wrdlen = 0;
      for( k = WorkAreaLedgeT7; k <= WorkAreaRedgeT7; k++ )
     {
          workbyte = xgamsCACHE[k];

                    if( wrdlen < MAXboth) {
                                        if (CaseSensitiveWildcardMatching_flag == 0)
                         wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                        else
                         wrdLOW[ wrdlen ] = ( workbyte );
                              wrd[ wrdlen ] = workbyte;
                      }
                    if (workbyte == 10) {
                              TotalLines7++;
// Wildcard search [
                    if ( 0 < wrdlen && wrdlen < MAXboth)
                      {
                              wrd[ wrdlen ] = 0;
                              wrdLOW[ wrdlen ] = 0;
                              if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                        {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                              WordsChecked7++;
                              if (Exact_flag) {
                                        //if ((long)( Railgun_Quadruplet_7Gulliver_7(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                        #ifdef RG7Gulliver
                                                  FOUNDinPTR = Railgun_Trolldom_7(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                  //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                                  //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_7(wrdLOW, wrdARG, wrdlen, n);
                                                  //FOUNDinPTR = Railgun_Sekireigan_Wolfram_7(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                        #else
                                                  FOUNDinPTR = Railgun_Quadruplet_7_7(wrdLOW, wrdARG, wrdlen, n);
                                        #endif
                                        // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                        // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                        //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                                        if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                        //if ((long)( Railgun_Quadruplet_7_7(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                  {
```

```
                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines7++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                  if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                                  }
                              } else {
                                        if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                        if ( WildcardMatch_Iterative_Kaze7(wrdARG, wrdLOW) ) {
#else
                                        if ( IterativeWildcards7(wrdARG, wrdLOW) ) {
#endif
                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines7++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                  if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                        }
                                        } else {
                                        maskGLOBALlen = n;
                                        nameGLOBALlen7 = wrdlen;
                                        if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_7(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines7++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                  if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                        }
                                        }
                              }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                    for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                    for (l=0; l < m-BB+1; l++) {
     WordsChecked7++;
// LD [
     SkipHeuristic=0;
     for(i=1;i<=BB;i++) {
          for (j=1;j<=n;j++) {
                    if(wrdLOW[l+i-1] == wrdARG[j-1])
                              LevenshteinT7[i][j] = LevenshteinT7[i-1][j-1];
```

```
        else
#if defined(_WIN32ASM_)
            LevenshteinT7[i][j] = min_AF(LevenshteinT7[i-1][j]+1, LevenshteinT7[i][j-1]+1, LevenshteinT7[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
            LevenshteinT7[i][j] = MIN(MIN((LevenshteinT7[i-1][j]+1),(LevenshteinT7[i][j-1]+1)),(LevenshteinT7[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
  jumpless code! // 358,327 xgrams/s
            //{LevenshteinT7[i][j] = MIN(MIN(LevenshteinT7[i-1][j],LevenshteinT7[i][j-1]),LevenshteinT7[i-1][j-1]); --LevenshteinT7[i][j];} // Variant 3: This compound line is
  much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
    }

    // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
    if ( LevenshteinT7[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT7[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
  nothing the logic is the same.
  }
            if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT7[BB][n]) {
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines7++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                goto EXHAUSTIVE7;
                //break; // No need of further checking down the line, one dump only is needed.
                //if ((DumpedLines & 0xff) == 0xff)
                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
            }
// LD ]
        }
        }
EXHAUSTIVE7:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

        for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
        for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
        // From here on 'm' will be replaced by BB
        if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
  WordsChecked7++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT7[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT7[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
  StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

  SkipHeuristic=0;
  for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
    for (j=1;j<=n;j++) {
        if(wrdLOW[l+i-1] == wrdARG[j-1])
            LevenshteinT7[i][j] = LevenshteinT7[i-1][j-1];
        else
#if defined(_WIN32ASM_)
            LevenshteinT7[i][j] = min_AF(LevenshteinT7[i-1][j]+1, LevenshteinT7[i][j-1]+1, LevenshteinT7[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
            LevenshteinT7[i][j] = MIN(MIN((LevenshteinT7[i-1][j]+1),(LevenshteinT7[i][j-1]+1)),(LevenshteinT7[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
  jumpless code! // 358,327 xgrams/s
            //{LevenshteinT7[i][j] = MIN(MIN(LevenshteinT7[i-1][j],LevenshteinT7[i][j-1]),LevenshteinT7[i-1][j-1]); --LevenshteinT7[i][j];} // Variant 3: This compound line is
  much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
    }

    // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
    if ( LevenshteinT7[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT7[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
  nothing the logic is the same.
  }
            if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT7[BB][n]) {
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines7++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                // Once dumping the line we need double 'break' from BB and l 'for's:
                goto EXHAUSTIVE7;
```

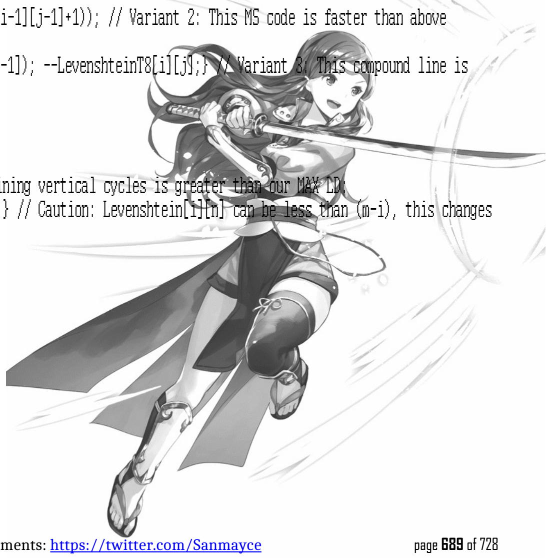


```
                                                        //if ((DumpedLines & 0xff) == 0xff)
                                                        //              //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                                        //              fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                          }
        // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
        //memcpy(wrdCACHEDT7, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
        memcpy(wrdCACHEDT7, &wrdLOW[l], BB); wrdCACHEDT7[BB]=0;
// LD ]
}

                            } // if (l < m-BB+1)
                            }
                            }
EXHAUSTIVE7:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
        WordsChecked7++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT7[StartingPosition-1] && wrdCACHEDT7[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
        StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

        SkipHeuristic=0;
        for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
              for (j=1;j<=n;j++) {
                            if(wrdLOW[i-1] == wrdARG[j-1])
                                          LevenshteinT7[i][j] = LevenshteinT7[i-1][j-1];
                            else
#if defined(_WIN32ASM_)
                                          LevenshteinT7[i][j] = min_AF(LevenshteinT7[i-1][j]+1, LevenshteinT7[i][j-1]+1, LevenshteinT7[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                                          LevenshteinT7[i][j] = MIN(MIN((LevenshteinT7[i-1][j]+1),(LevenshteinT7[i][j-1]+1)),(LevenshteinT7[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
        jumpless code! // 358,327 xgrams/s
                                          //{LevenshteinT7[i][j] = MIN(MIN(LevenshteinT7[i-1][j],LevenshteinT7[i][j-1]),LevenshteinT7[i-1][j-1]); --LevenshteinT7[i][j];} // Variant 3: This compound line is
        much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
              }

              // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
              if ( LevenshteinT7[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT7[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
        nothing the logic is the same.
        }
                                          if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT7[m][n]) {
                                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines7++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                        #pragma omp critical
#endif
                                                        {
                                                        fprintf( fp_outLINE, "%s\r\n", wrd);
                                                        }
}
                                                        //if ((DumpedLines & 0xff) == 0xff)
                                                        //              //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                                        //              fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                          }
        // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
        memcpy(wrdCACHEDT7, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                            }
// Wildcard search ]
                                          wrdlen = 0;
                         }
                         else wrdlen++;
         } // k 'for'
         //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                          } // if (Exact_flag) {
        }

// 8th thread
#ifdef Commence_OpenMP
    #pragma omp section
#endif
        {
//This fragment is bugggy (somehow) [
```

```
                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[[ Since r.1-++
                                k = WorkAreaLedgeT8;
                                while ( k < WorkAreaRedgeT8 ) {
                                #ifdef RG7Gulliver
                                        FOUNDinPTR = Railgun_Trolldom_8(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT8-k+1, n); // 2019-May-17
                                        //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_8(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT8-k+1, n);
                                        //FOUNDinPTR = Railgun_Sekireigan_Wolfram_8(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT8-k+1, n); // 2019-May-17
                                #else
                                        FOUNDinPTR = Railgun_Quadruplet_7_8(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT8-k+1, n);
                                #endif
                                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                //if ((long)( FOUNDinPTR - &xgamsCACHE[k] ))>=0) {
                                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                        i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                        while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                        while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                        k = j+1; // Should "point" to first symbol after the dumped fragment.

                                        //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                        if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                        memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines8++;
                                        j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                        j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                        if (Dump_flag)
                                        fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                        else
                                        fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                        }
                                } else k = WorkAreaRedgeT8;
                                } // while
// WHOLE buffer at once not line-by-line ]]]]]]]] Since r.1-++
                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                        //{

    //memset (wrdCACHEDT8, 0, MaxLineLength+1+1);
    i = 1;
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    wrdlen = 0;
     for( k = WorkAreaLedgeT8; k <= WorkAreaRedgeT8; k++ )
    {
        workbyte = xgamsCACHE[k];

              if( wrdlen < MAXboth) {
                                if (CaseSensitiveWildcardMatching_flag == 0)
                  wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                else
                  wrdLOW[ wrdlen ] = ( workbyte );
                        wrd[ wrdlen ] = workbyte;
                }
              if (workbyte == 10) {
                        TotalLines8++;
// Wildcard search [
              if ( 0 < wrdlen && wrdlen < MAXboth)
                {
                        wrd[ wrdlen ] = 0;
                        wrdLOW[ wrdlen ] = 0;
                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}


if (WILDCARD_IP_flag) {
// WILDCARD IP [
                        WordsChecked8++;
                        if (Exact_flag) {
                                //if ((long)( Railgun_Quadruplet_7Gulliver_8(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))>=0)
                                #ifdef RG7Gulliver
                                        FOUNDinPTR = Railgun_Trolldom_8(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                        //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                        //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_8(wrdLOW, wrdARG, wrdlen, n);
                                        //FOUNDinPTR = Railgun_Sekireigan_Wolfram_8(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                #else
                                        FOUNDinPTR = Railgun_Quadruplet_7_8(wrdLOW, wrdARG, wrdlen, n);
                                #endif
                                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                //if ((long)( FOUNDinPTR - wrdLOW ))>=0)
                                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
```

```
                                                        //if ((long)( Railgun_Quadruplet_7_8(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))>=0)
                                                                      {
                                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines8++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                        #pragma omp critical
#endif
                                                        {
                                                                      if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                        }
}
                                                                      }
                                          } else {
                                                        if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                                        if ( WildcardMatch_Iterative_Kaze8(wrdARG, wrdLOW) ) {
#else
                                                        if ( IterativeWildcards8(wrdARG, wrdLOW) ) {
#endif
                                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines8++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                        #pragma omp critical
#endif
                                                        {
                                                                      if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                        }
}
                                                        }
                                                        } else {
                                                        maskGLOBALlen = n;
                                                        nameGLOBALlen8 = wrdlen;
                                                        if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_8(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines8++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                        #pragma omp critical
#endif
                                                        {
                                                                      if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                        }
}
                                                        }
                                                        }
                                          }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                            for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                            for (l=0; l < m-BB+1; l++) {
       WordsChecked8++;
// LD [
       SkipHeuristic=0;
       for(i=1;i<=BB;i++) {
             for (j=1;j<=n;j++) {
```

```
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT8[i][j] = LevenshteinT8[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT8[i][j] = min_AF(LevenshteinT8[i-1][j]+1, LevenshteinT8[i][j-1]+1, LevenshteinT8[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT8[i][j] = MIN(MIN((LevenshteinT8[i-1][j]+1),(LevenshteinT8[i][j-1]+1)),(LevenshteinT8[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT8[i][j] = MIN(MIN(LevenshteinT8[i-1][j],LevenshteinT8[i][j-1]),LevenshteinT8[i-1][j-1]); --LevenshteinT8[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT8[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT8[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT8[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines8++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVE8;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE8:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked8++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT8[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT8[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT8[i][j] = LevenshteinT8[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT8[i][j] = min_AF(LevenshteinT8[i-1][j]+1, LevenshteinT8[i][j-1]+1, LevenshteinT8[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT8[i][j] = MIN(MIN((LevenshteinT8[i-1][j]+1),(LevenshteinT8[i][j-1]+1)),(LevenshteinT8[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT8[i][j] = MIN(MIN(LevenshteinT8[i-1][j],LevenshteinT8[i][j-1]),LevenshteinT8[i-1][j-1]); --LevenshteinT8[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT8[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT8[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT8[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines8++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
```

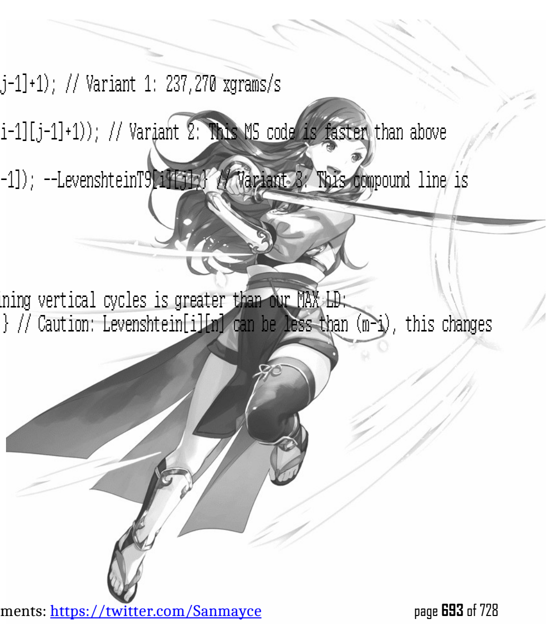


```
                                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                                goto EXHAUSTIVE8;
                                                //if ((DumpedLines & 0xff) == 0xff)
                                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                    }
        // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
        //memcpy(wrdCACHEDT8, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
        memcpy(wrdCACHEDT8, &wrdLOW[l], BB); wrdCACHEDT8[BB]=0;
// LD ]
}

                        } // if (l < m-BB+1)
                        }
                        }
EXHAUSTIVE8:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
        WordsChecked8++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT8[StartingPosition-1] && wrdCACHEDT8[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
        StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

        SkipHeuristic=0;
        for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
              for (j=1;j<=n;j++) {
                        if(wrdLOW[i-1] == wrdARG[j-1])
                                    LevenshteinT8[i][j] = LevenshteinT8[i-1][j-1];
                        else
#if defined(_WIN32ASM_)
                                    LevenshteinT8[i][j] = min_AF(LevenshteinT8[i-1][j]+1, LevenshteinT8[i][j-1]+1, LevenshteinT8[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                                    LevenshteinT8[i][j] = MIN(MIN((LevenshteinT8[i-1][j]+1),(LevenshteinT8[i][j-1]+1)),(LevenshteinT8[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
        jumpless code! // 358,327 xgrams/s
                                    //{LevenshteinT8[i][j] = MIN(MIN(LevenshteinT8[i-1][j],LevenshteinT8[i][j-1]),LevenshteinT8[i-1][j-1]); --LevenshteinT8[i][j];} // Variant 3: This compound line is
        much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
              }

              // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
              if ( LevenshteinT8[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT8[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
        nothing the logic is the same.
        }
                                    if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT8[m][n]) {
                                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines8++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                #pragma omp critical

#endif
                                                {
                                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                                }
}
                                                //if ((DumpedLines & 0xff) == 0xff)
                                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                    }
        // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
        memcpy(wrdCACHEDT8, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                        }
// Wildcard search ]
                                    wrdlen = 0;
                      }
                      else wrdlen++;
        } // k 'for'
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                    } // if (Exact_flag) {
        }

// 9th thread
#ifdef Commence_OpenMP
   #pragma omp section
#endif
```

```
       {
//This fragment is bugggy (somehow) [

                                       if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                                    k = WorkAreaLedgeT9;
                                                    while ( k < WorkAreaRedgeT9 ) {
                                                    #ifdef RG7Gulliver
                                                                 FOUNDinPTR = Railgun_Trolldom_9(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT9-k+1, n); // 2019-May-17
                                                                 //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_9(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT9-k+1, n);
                                                                 //FOUNDinPTR = Railgun_Sekireigan_Wolfram_9(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT9-k+1, n); // 2019-May-17
                                                    #else
                                                                 FOUNDinPTR = Railgun_Quadruplet_7_9(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT9-k+1, n);
                                                    #endif
                                                    // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                    // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                    //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                                    if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                 i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                 while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                 while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                 k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                 //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                 if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                 memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines9++;
                                                                 j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                 j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                    // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                    #pragma omp critical
#endif
                                                    {
                                                                 if (Dump_flag)
                                                                 fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                 else
                                                                 fprintf( fp_outLINE, "%s\r\n", wrd);
                                                    }
}
                                                                 }
                                                    } else k = WorkAreaRedgeT9;
                                                    } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                                       } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                       //{

       //memset (wrdCACHEDT9, 0, MaxLineLength+1+1);
       i = 1;
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       wrdlen = 0;
        for( k = WorkAreaLedgeT9; k <= WorkAreaRedgeT9; k++ )
       {
             workbyte = xgamsCACHE[k];

                       if( wrdlen < MAXboth) {
                                                    if (CaseSensitiveWildcardMatching_flag == 0)
                             wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                    else
                             wrdLOW[ wrdlen ] = ( workbyte );
                                       wrd[ wrdlen ] = workbyte;
                         }
                       if (workbyte == 10) {
                                       TotalLines9++;
// Wildcard search [
                       if ( 0 < wrdlen && wrdlen < MAXboth)
                         {
                                       wrd[ wrdlen ] = 0;
                                       wrdLOW[ wrdlen ] = 0;
                                       if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                    {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                       WordsChecked9++;
                                       if (Exact_flag) {
                                                    //if ((long)( Railgun_Quadruplet_7Gulliver_9(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                    #ifdef RG7Gulliver
                                                                 FOUNDinPTR = Railgun_Trolldom_9(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                                 //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                                                 //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_9(wrdLOW, wrdARG, wrdlen, n);
                                                                 //FOUNDinPTR = Railgun_Sekireigan_Wolfram_9(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                    #else
                                                                 FOUNDinPTR = Railgun_Quadruplet_7_9(wrdLOW, wrdARG, wrdlen, n);
                                                    #endif
                                                    // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                    // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
```

```
                                //if ((long)( FOUNDinPTR - wrdLOW ))>=0)
                                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                //if ((long)( Railgun_Quadruplet_7_9(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))>=0)
                                        {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines9++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                        if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                        }
                        } else {
                                if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                if ( WildcardMatch_Iterative_Kaze9(wrdARG, wrdLOW) ) {
#else
                                if ( IterativeWildcards9(wrdARG, wrdLOW) ) {
#endif
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines9++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                        if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                }
                                } else {
                                maskGLOBALlen = n;
                                nameGLOBALlen9 = wrdlen;
                                if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_9(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines9++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                        if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                }
                                }
                        }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                for (l=0; l < m-BB+1; l++) {
    WordsChecked9++;
// LD [
    SkipHeuristic=0;
```

```
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT9[i][j] = LevenshteinT9[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT9[i][j] = min_AF(LevenshteinT9[i-1][j]+1, LevenshteinT9[i][j-1]+1, LevenshteinT9[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT9[i][j] = MIN(MIN((LevenshteinT9[i-1][j]+1),(LevenshteinT9[i][j-1]+1)),(LevenshteinT9[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT9[i][j] = MIN(MIN(LevenshteinT9[i-1][j],LevenshteinT9[i][j-1]),LevenshteinT9[i-1][j-1]); --LevenshteinT9[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT9[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT9[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT9[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines9++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVE9;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE9:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked9++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT9[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT9[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT9[i][j] = LevenshteinT9[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT9[i][j] = min_AF(LevenshteinT9[i-1][j]+1, LevenshteinT9[i][j-1]+1, LevenshteinT9[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT9[i][j] = MIN(MIN((LevenshteinT9[i-1][j]+1),(LevenshteinT9[i][j-1]+1)),(LevenshteinT9[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT9[i][j] = MIN(MIN(LevenshteinT9[i-1][j],LevenshteinT9[i][j-1]),LevenshteinT9[i-1][j-1]); --LevenshteinT9[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT9[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT9[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT9[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines9++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
```

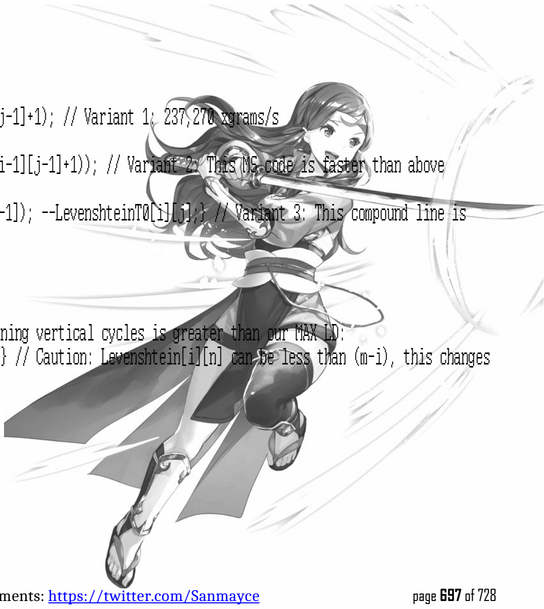


```
                                    }
}
                                    // Once dumping the line we need double 'break' from BB and 1 'for's:
                                    goto EXHAUSTIVE9;
                                    //if ((DumpedLines & 0xff) == 0xff)
                                    //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                    //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                            }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDT9, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDT9, &wrdLOW[1], BB); wrdCACHEDT9[BB]=0;
// LD ]
}

                } // if (1 < m-BB+1)
                }
                }
EXHAUSTIVE9:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsChecked9++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT9[StartingPosition-1] && wrdCACHEDT9[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinT9[i][j] = LevenshteinT9[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT9[i][j] = min_AF(LevenshteinT9[i-1][j]+1, LevenshteinT9[i][j-1]+1, LevenshteinT9[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT9[i][j] = MIN(MIN((LevenshteinT9[i-1][j]+1),(LevenshteinT9[i][j-1]+1)),(LevenshteinT9[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT9[i][j] = MIN(MIN(LevenshteinT9[i-1][j],LevenshteinT9[i][j-1]),LevenshteinT9[i-1][j-1]); --LevenshteinT9[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT9[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT9[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT9[m][n]) {
                                    // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines9++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                    #pragma omp critical
#endif
                                    {
                                    fprintf( fp_outLINE, "%s\r\n", wrd);
                                    }
}
                                    //if ((DumpedLines & 0xff) == 0xff)
                                    //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                    //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDT9, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
               }
               else wrdlen++;
     } // k 'for'
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        } // if (Exact_flag) {
    }

// 0th thread
#ifdef Commence_OpenMP
```

```
   #pragma omp section
#endif
     {
//This fragment is bugggy (somehow) [

                            if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[[ Since r.1-++
                                     k = WorkAreaLedgeT0;
                                     while ( k < WorkAreaRedgeT0 ) {
                                     #ifdef RG7Gulliver
                                               FOUNDinPTR = Railgun_Trolldom_0(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT0-k+1, n); // 2019-May-17
                                               //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_0(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT0-k+1, n);
                                               //FOUNDinPTR = Railgun_Sekireigan_Wolfram_0(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT0-k+1, n); // 2019-May-17
                                     #else
                                               FOUNDinPTR = Railgun_Quadruplet_7_0(&xgamsCACHE[k], wrdARG, WorkAreaRedgeT0-k+1, n);
                                     #endif
                                     // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                     // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                     //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                     if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                               i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                               while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                               while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                               k = j+1; // Should "point" to first symbol after the dumped fragment.

                                               //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                               if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                               memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines0++;
                                               j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                               j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                     // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP

                                     #pragma omp critical

#endif

                                     {
                                               if (Dump_flag)
                                               fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                               else
                                               fprintf( fp_outLINE, "%s\r\n", wrd);
                                     }
}

                                               }
                                     } else k = WorkAreaRedgeT0;
                                     } // while
// WHOLE buffer at once not line-by-line ]]]]]]]] Since r.1-++
                            } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                            //{

     //memset (wrdCACHEDT0, 0, MaxLineLength+1+1);
     i = 1;
      //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     wrdlen = 0;
      for( k = WorkAreaLedgeT0; k <= WorkAreaRedgeT0; k++ )
     {
         workbyte = xgamsCACHE[k];

                 if( wrdlen < MAXboth) {
                                     if (CaseSensitiveWildcardMatching_flag == 0)
                      wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                     else
                      wrdLOW[ wrdlen ] = ( workbyte );
                            wrd[ wrdlen ] = workbyte;
                   }
                 if (workbyte == 10) {
                            TotalLines0++;
// Wildcard search [
                 if ( 0 < wrdlen && wrdlen < MAXboth)
                   {
                            wrd[ wrdlen ] = 0;
                            wrdLOW[ wrdlen ] = 0;
                            if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                     {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                            WordsChecked0++;
                            if (Exact_flag) {
                                     //if ((long)( Railgun_Quadruplet_7Gulliver_0(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                     #ifdef RG7Gulliver
                                               FOUNDinPTR = Railgun_Trolldom_0(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                               //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                               //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_0(wrdLOW, wrdARG, wrdlen, n);
                                               //FOUNDinPTR = Railgun_Sekireigan_Wolfram_0(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                     #else
                                               FOUNDinPTR = Railgun_Quadruplet_7_0(wrdLOW, wrdARG, wrdlen, n);
                                     #endif
```

```
                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                //if ((long)( FOUNDinPTR - wrdLOW ))=0)
                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                //if ((long)( Railgun_Quadruplet_7_0(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))=0)
                    {
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines0++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                    }
            } else {
                if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                if ( WildcardMatch_Iterative_Kaze0(wrdARG, wrdLOW) ) {
#else
                if ( IterativeWildcards0(wrdARG, wrdLOW) ) {
#endif
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines0++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                } else {
                maskGLOBALlen = n;
                nameGLOBALlen0 = wrdlen;
                if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_0(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines0++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                }
            }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// |Edelweiß || [[edelweiss]] || edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// |Edelweiß |
// Edelweiß ||
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
        for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
        for (l=0; l < m-BB+1; l++) {
  WordsChecked0++;
```

```
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT0[i][j] = LevenshteinT0[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT0[i][j] = min_AF(LevenshteinT0[i-1][j]+1, LevenshteinT0[i][j-1]+1, LevenshteinT0[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT0[i][j] = MIN(MIN((LevenshteinT0[i-1][j]+1),(LevenshteinT0[i][j-1]+1)),(LevenshteinT0[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT0[i][j] = MIN(MIN(LevenshteinT0[i-1][j],LevenshteinT0[i][j-1]),LevenshteinT0[i-1][j-1]); --LevenshteinT0[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT0[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT0[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT0[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines0++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVE0;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE0:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked0++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT0[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT0[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT0[i][j] = LevenshteinT0[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT0[i][j] = min_AF(LevenshteinT0[i-1][j]+1, LevenshteinT0[i][j-1]+1, LevenshteinT0[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT0[i][j] = MIN(MIN((LevenshteinT0[i-1][j]+1),(LevenshteinT0[i][j-1]+1)),(LevenshteinT0[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT0[i][j] = MIN(MIN(LevenshteinT0[i-1][j],LevenshteinT0[i][j-1]),LevenshteinT0[i-1][j-1]); --LevenshteinT0[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT0[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT0[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT0[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines0++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
```

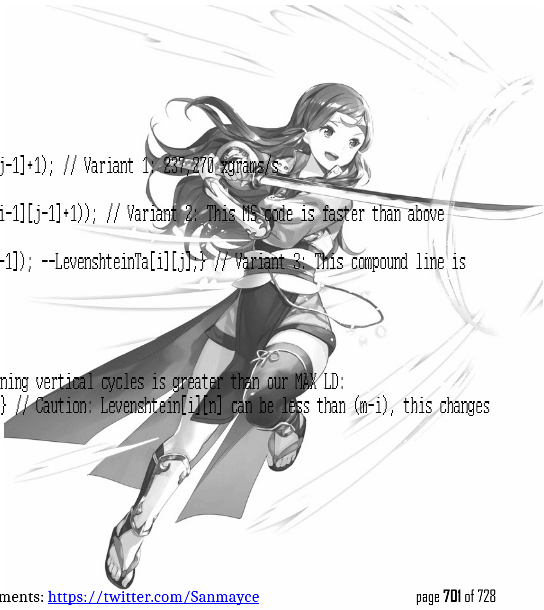


```
                                    {
                                    fprintf( fp_outLINE, "%s\r\n", wrd);
                                    }
}
                                    // Once dumping the line we need double 'break' from BB and l 'for's:
                                    goto EXHAUSTIVE0;
                                    //if ((DumpedLines & 0xff) == 0xff)
                                    //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                    //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                            }
        // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
        //memcpy(wrdCACHEDT0, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
        memcpy(wrdCACHEDT0, &wrdLOW[l], BB); wrdCACHEDT0[BB]=0;
// LD ]
}

                    } // if (l < m-BB+1)
                    }
                    }
EXHAUSTIVE0:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
        WordsChecked0++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT0[StartingPosition-1] && wrdCACHEDT0[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
        StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

        SkipHeuristic=0;
        for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
            for (j=1;j<=n;j++) {
                    if(wrdLOW[i-1] == wrdARG[j-1])
                            LevenshteinT0[i][j] = LevenshteinT0[i-1][j-1];
                    else
#if defined(_WIN32ASM_)
                            LevenshteinT0[i][j] = min_AF(LevenshteinT0[i-1][j]+1, LevenshteinT0[i][j-1]+1, LevenshteinT0[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                            LevenshteinT0[i][j] = MIN(MIN((LevenshteinT0[i-1][j]+1),(LevenshteinT0[i][j-1]+1)),(LevenshteinT0[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
        jumpless code! // 358,327 xgrams/s
                            //{LevenshteinT0[i][j] = MIN(MIN(LevenshteinT0[i-1][j],LevenshteinT0[i][j-1]),LevenshteinT0[i-1][j-1]); --LevenshteinT0[i][j];} // Variant 3: This compound line is
        much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
            }

            // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
            if ( LevenshteinT0[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT0[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
        nothing the logic is the same.
        }
                            if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT0[m][n]) {
                                    // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLines0++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                    #pragma omp critical
#endif
                                    {
                                    fprintf( fp_outLINE, "%s\r\n", wrd);
                                    }
}
                                    //if ((DumpedLines & 0xff) == 0xff)
                                    //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                    //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                            }
        // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
        memcpy(wrdCACHEDT0, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                    }
// Wildcard search ]
                            wrdlen = 0;
                  }
                  else wrdlen++;
        } // k 'for'
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                            } // if (Exact_flag) {
        }
```

```
// Ath thread
#ifdef Commence_OpenMP
    #pragma omp section
#endif
       {
//This fragment is bugggy (somehow) [

                                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                                     k = WorkAreaLedgeTa;
                                                     while ( k < WorkAreaRedgeTa ) {
                                                     #ifdef RG7Gulliver
                                                                  FOUNDinPTR = Railgun_Trolldom_a(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTa-k+1, n); // 2019-May-17
                                                                  //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_a(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTa-k+1, n);
                                                                  //FOUNDinPTR = Railgun_Sekireigan_Wolfram_a(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTa-k+1, n); // 2019-May-17
                                                     #else
                                                                  FOUNDinPTR = Railgun_Quadruplet_7_a(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTa-k+1, n);
                                                     #endif
                                                     // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                     // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                     //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                                     if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                  i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                  while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                  while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                  k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                  //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                  if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                  memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLinesa++;
                                                                  j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                  j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                     // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                     #pragma omp critical

#endif
                                                     {
                                                                  if (Dump_flag)
                                                                  fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                  else
                                                                  fprintf( fp_outLINE, "%s\r\n", wrd);
                                                     }
}
                                                                  }
                                                     } else k = WorkAreaRedgeTa;
                                                     } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                        //{


       //memset (wrdCACHEDTa, 0, MaxLineLength+1+1);
       i = 1;
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       wrdlen = 0;
        for( k = WorkAreaLedgeTa; k <= WorkAreaRedgeTa; k++ )
       {
             workbyte = xgamsCACHE[k];

                        if( wrdlen < MAXboth) {
                                                     if (CaseSensitiveWildcardMatching_flag == 0)
                              wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                     else
                              wrdLOW[ wrdlen ] = ( workbyte );
                                        wrd[ wrdlen ] = workbyte;
                          }
                        if (workbyte == 10) {
                                        TotalLinesa++;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                          {
                                        wrd[ wrdlen ] = 0;
                                        wrdLOW[ wrdlen ] = 0;
                                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                     {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                        WordsCheckeda++;
                                        if (Exact_flag) {
                                                     //if ((long)( Railgun_Quadruplet_7Gulliver_a(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                     #ifdef RG7Gulliver
                                                                  FOUNDinPTR = Railgun_Trolldom_a(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                                  //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                                                  //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_a(wrdLOW, wrdARG, wrdlen, n);
                                                                  //FOUNDinPTR = Railgun_Sekireigan_Wolfram_a(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                     #else
```

```
                                                                   FOUNDinPTR = Railgun_Quadruplet_7_a(wrdLOW, wrdARG, wrdlen, n);
                                                      #endif
                                                      // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                      // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                      //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                                                      if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                                      //if ((long)( Railgun_Quadruplet_7_a(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                                   {
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesa++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                                   }
                                        } else {
                                                      if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                                      if ( WildcardMatch_Iterative_Kazea(wrdARG, wrdLOW) ) {
#else
                                                      if ( IterativeWildcardsa(wrdARG, wrdLOW) ) {
#endif
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesa++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                      }
                                                      } else {
                                                      maskGLOBALlen = n;
                                                      nameGLOBALlena = wrdlen;
                                                      if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_a(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesa++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical
#endif
                                                      {
                                                                   if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                      }
                                                      }
                                        }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// |Edelweiß || [[edelweiss]] || edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// |Edelweiß |
// Edelweiß ||
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                           for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
```

```
                for (l=0; l < m-BB+1; l++) {
    WordsCheckeda++;
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTa[i][j] = LevenshteinTa[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTa[i][j] = min_AF(LevenshteinTa[i-1][j]+1, LevenshteinTa[i][j-1]+1, LevenshteinTa[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTa[i][j] = MIN(MIN((LevenshteinTa[i-1][j]+1),(LevenshteinTa[i][j-1]+1)),(LevenshteinTa[i-1][j-1]+1)); // Variant 2: This MS code is faster than above jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTa[i][j] = MIN(MIN(LevenshteinTa[i-1][j],LevenshteinTa[i][j-1]),LevenshteinTa[i-1][j-1]); --LevenshteinTa[i][j];} // Variant 3: This compound line is much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTa[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTa[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTa[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesa++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVEa;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVEa:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsCheckeda++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTa[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDTa[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTa[i][j] = LevenshteinTa[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTa[i][j] = min_AF(LevenshteinTa[i-1][j]+1, LevenshteinTa[i][j-1]+1, LevenshteinTa[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTa[i][j] = MIN(MIN((LevenshteinTa[i-1][j]+1),(LevenshteinTa[i][j-1]+1)),(LevenshteinTa[i-1][j-1]+1)); // Variant 2: This MS code is faster than above jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTa[i][j] = MIN(MIN(LevenshteinTa[i-1][j],LevenshteinTa[i][j-1]),LevenshteinTa[i-1][j-1]); --LevenshteinTa[i][j];} // Variant 3: This compound line is much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTa[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTa[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTa[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesa++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
```

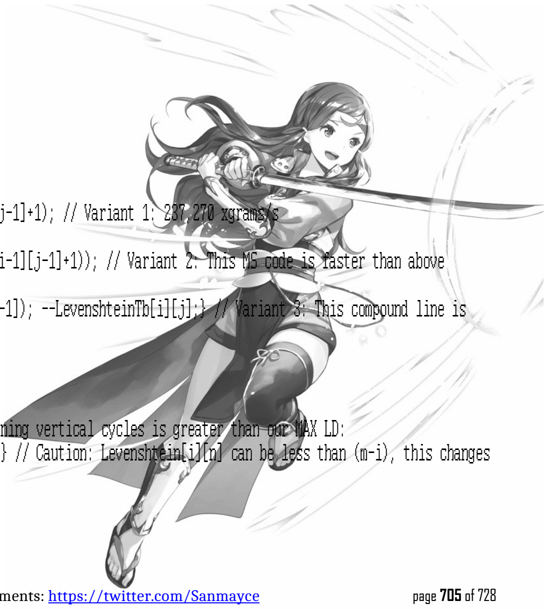


```
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVEa;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDTa, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDTa, &wrdLOW[l], BB); wrdCACHEDTa[BB]=0;
// LD ]
}

                } // if (l < m-BB+1)
                }
                }
EXHAUSTIVEa:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsCheckeda++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTa[StartingPosition-1] && wrdCACHEDTa[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinTa[i][j] = LevenshteinTa[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTa[i][j] = min_AF(LevenshteinTa[i-1][j]+1, LevenshteinTa[i][j-1]+1, LevenshteinTa[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTa[i][j] = MIN(MIN((LevenshteinTa[i-1][j]+1),(LevenshteinTa[i][j-1]+1)),(LevenshteinTa[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTa[i][j] = MIN(MIN(LevenshteinTa[i-1][j],LevenshteinTa[i][j-1]),LevenshteinTa[i-1][j-1]); --LevenshteinTa[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTa[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTa[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTa[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesa++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDTa, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
              }
              else wrdlen++;
     } // k 'for'
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        } // if (Exact_flag) {
```

```
       }

// Bth thread
#ifdef Commence_OpenMP
    #pragma omp section
#endif
       {
//This fragment is bugggy (somehow) [

                                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[[ Since r.1-++
                                                     k = WorkAreaLedgeTb;
                                                     while ( k < WorkAreaRedgeTb ) {
                                                     #ifdef RG7Gulliver
                                                                   FOUNDinPTR = Railgun_Trolldom_b(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTb-k+1, n); // 2019-May-17
                                                                   //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_b(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTb-k+1, n);
                                                                   //FOUNDinPTR = Railgun_Sekireigan_Wolfram_b(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTb-k+1, n); // 2019-May-17
                                                     #else
                                                                   FOUNDinPTR = Railgun_Quadruplet_7_b(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTb-k+1, n);
                                                     #endif
                                                     // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                     // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                     //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                                     if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                   i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                   while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                   while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                   k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                   //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                   if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                   memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLinesb++;
                                                                   j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                   j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                     // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                     #pragma omp critical

#endif
                                                     {
                                                                   if (Dump_flag)
                                                                   fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                   else
                                                                   fprintf( fp_outLINE, "%s\r\n", wrd);
                                                     }
}
                                                                   }
                                                     } else k = WorkAreaRedgeTb;
                                                     } // while
// WHOLE buffer at once not line-by-line ]]]]]]]] Since r.1-++
                                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                        //{

       //memset (wrdCACHEDTb, 0, MaxLineLength+1+1);
       i = 1;
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       wrdlen = 0;
        for( k = WorkAreaLedgeTb; k <= WorkAreaRedgeTb; k++ )
       {
             workbyte = xgamsCACHE[k];

                        if( wrdlen < MAXboth) {
                                                     if (CaseSensitiveWildcardMatching_flag == 0)
                               wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                     else
                               wrdLOW[ wrdlen ] = ( workbyte );
                                        wrd[ wrdlen ] = workbyte;
                           }
                        if (workbyte == 10) {
                                        TotalLinesb++;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                           {
                                        wrd[ wrdlen ] = 0;
                                        wrdLOW[ wrdlen ] = 0;
                                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                     {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                        WordsCheckedb++;
                                        if (Exact_flag) {
                                                     //if ((long)( Railgun_Quadruplet_7Gulliver_b(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                     #ifdef RG7Gulliver
                                                                   FOUNDinPTR = Railgun_Trolldom_b(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                                   //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                                                   //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_b(wrdLOW, wrdARG, wrdlen, n);
```

```
                                                //FOUNDinPTR = Railgun_Sekireigan_Wolfram_b(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                        #else
                                                FOUNDinPTR = Railgun_Quadruplet_7_b(wrdLOW, wrdARG, wrdlen, n);
                                        #endif
                                        // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                        // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                        //if ((long)( FOUNDinPTR - wrdLOW ))>=0)
                                        if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                        //if ((long)( Railgun_Quadruplet_7_b(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))>=0)
                                                {
                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesb++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                                }
                                } else {
                                        if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                        if ( WildcardMatch_Iterative_Kazeb(wrdARG, wrdLOW) ) {
#else
                                        if ( IterativeWildcardsb(wrdARG, wrdLOW) ) {
#endif
                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesb++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                        }
                                        } else {
                                        maskGLOBALlen = n;
                                        nameGLOBALlenb = wrdlen;
                                        if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_b(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                        // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesb++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                        #pragma omp critical
#endif
                                        {
                                                if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                                        }
}
                                        }
                                        }
                                }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

```
/*
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                for (l=0; l < m-BB+1; l++) {
    WordsCheckedb++;
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTb[i][j] = LevenshteinTb[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTb[i][j] = min_AF(LevenshteinTb[i-1][j]+1, LevenshteinTb[i][j-1]+1, LevenshteinTb[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTb[i][j] = MIN(MIN((LevenshteinTb[i-1][j]+1),(LevenshteinTb[i][j-1]+1)),(LevenshteinTb[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTb[i][j] = MIN(MIN(LevenshteinTb[i-1][j],LevenshteinTb[i][j-1]),LevenshteinTb[i-1][j-1]); --LevenshteinTb[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTb[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTb[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTb[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesb++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVEb;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVEb:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsCheckedb++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTb[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDTb[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTb[i][j] = LevenshteinTb[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTb[i][j] = min_AF(LevenshteinTb[i-1][j]+1, LevenshteinTb[i][j-1]+1, LevenshteinTb[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTb[i][j] = MIN(MIN((LevenshteinTb[i-1][j]+1),(LevenshteinTb[i][j-1]+1)),(LevenshteinTb[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTb[i][j] = MIN(MIN(LevenshteinTb[i-1][j],LevenshteinTb[i][j-1]),LevenshteinTb[i-1][j-1]); --LevenshteinTb[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTb[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTb[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTb[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesb++;
```

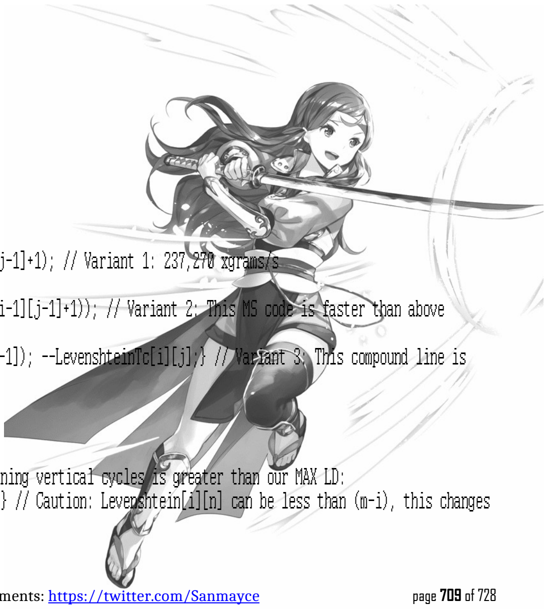


```
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                #pragma omp critical
#endif
                                                {
                                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                                }
}
                                                // Once dumping the line we need double ‘break’ from BB and 1 ‘for’s:
                                                goto EXHAUSTIVEb;
                                                //if ((DumpedLines & 0xff) == 0xff)
                                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                    }
        // The bail out ‘i’ value (heuristic #2) affects our cached value here, ‘i’ is the needed one:
        //memcpy(wrdCACHEDTb, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
        memcpy(wrdCACHEDTb, &wrdLOW[1], BB); wrdCACHEDTb[BB]=0;
// LD ]
}

                        } // if (1 < m-BB+1)
                        }
                        }
EXHAUSTIVEb:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
        WordsCheckedb++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTb[StartingPosition-1] && wrdCACHEDTb[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
        StartingPosition++;
// The bail out ‘i’ value (heuristic #2) affects our cached value here, ‘StartingPosition’ cannot be greater than ‘i’:
StartingPosition = MIN(StartingPosition, i);

        SkipHeuristic=0;
        for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
            for (j=1;j<=n;j++) {
                        if(wrdLOW[i-1] == wrdARG[j-1])
                                    LevenshteinTb[i][j] = LevenshteinTb[i-1][j-1];
                        else
#if defined(_WIN32ASM_)
                                    LevenshteinTb[i][j] = min_AF(LevenshteinTb[i-1][j]+1, LevenshteinTb[i][j-1]+1, LevenshteinTb[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                                    LevenshteinTb[i][j] = MIN(MIN((LevenshteinTb[i-1][j]+1),(LevenshteinTb[i][j-1]+1)),(LevenshteinTb[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
        jumpless code! // 358,327 xgrams/s
                                    //{LevenshteinTb[i][j] = MIN(MIN(LevenshteinTb[i-1][j],LevenshteinTb[i][j-1]),LevenshteinTb[i-1][j-1]); --LevenshteinTb[i][j];} // Variant 3: This compound line is
        much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
            }

            // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
            if ( LevenshteinTb[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTb[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
        nothing the logic is the same.
        }
                                    if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTb[m][n]) {
                                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesb++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                #pragma omp critical
#endif
                                                {
                                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                                }
}
                                                //if ((DumpedLines & 0xff) == 0xff)
                                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                    }
        // The bail out ‘i’ value (heuristic #2) affects our cached value here, ‘i’ is the needed one:
        memcpy(wrdCACHEDTb, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                        }
// Wildcard search ]
                                    wrdlen = 0;
                      }
                      else wrdlen++;
         } // k 'for'
```

```
         //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                        } // if (Exact_flag) {
        }

// Cth thread
#ifdef Commence_OpenMP
    #pragma omp section
#endif
        {
//This fragment is bugggy (somehow) [

                                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                                    k = WorkAreaLedgeTc;
                                                    while ( k < WorkAreaRedgeTc ) {
                                                    #ifdef RG7Gulliver
                                                                FOUNDinPTR = Railgun_Trolldom_c(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTc-k+1, n); // 2019-May-17
                                                                //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_c(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTc-k+1, n);
                                                                //FOUNDinPTR = Railgun_Sekireigan_Wolfram_c(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTc-k+1, n); // 2019-May-17
                                                    #else
                                                                FOUNDinPTR = Railgun_Quadruplet_7_c(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTc-k+1, n);
                                                    #endif
                                                    // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                    // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                    //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                                    if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLinesc++;
                                                                j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                    // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP

                                                    #pragma omp critical
#endif
                                                    {
                                                                if (Dump_flag)
                                                                fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                else
                                                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                                    }
}
                                                                }
                                                    } else k = WorkAreaRedgeTc;
                                                    } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                        //{

        //memset (wrdCACHEDTc, 0, MaxLineLength+1+1);
        i = 1;
         //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        wrdlen = 0;
         for( k = WorkAreaLedgeTc; k <= WorkAreaRedgeTc; k++ )
        {
              workbyte = xgamsCACHE[k];

                        if( wrdlen < MAXboth) {
                                                    if (CaseSensitiveWildcardMatching_flag == 0)
                               wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                    else
                               wrdLOW[ wrdlen ] = ( workbyte );
                                        wrd[ wrdlen ] = workbyte;
                           }
                        if (workbyte == 10) {
                                        TotalLinesc++;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                           {
                                        wrd[ wrdlen ] = 0;
                                        wrdLOW[ wrdlen ] = 0;
                                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                    {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                        WordsCheckedc++;
                                        if (Exact_flag) {
                                                    //if ((long)( Railgun_Quadruplet_7Gulliver_c(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                    #ifdef RG7Gulliver
                                                                FOUNDinPTR = Railgun_Trolldom_c(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
```

```
                    //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                    //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_c(wrdLOW, wrdARG, wrdlen, n);
                    //FOUNDinPTR = Railgun_Sekireigan_Wolfram_c(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                #else
                    FOUNDinPTR = Railgun_Quadruplet_7_c(wrdLOW, wrdARG, wrdlen, n);
                #endif
                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                //if ((long)( FOUNDinPTR - wrdLOW ))>=0)
                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                //if ((long)( Railgun_Quadruplet_7_c(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))>=0)
                    {
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesc++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                    }
            } else {
                if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                if ( WildcardMatch_Iterative_Kazec(wrdARG, wrdLOW) ) {
#else
                if ( IterativeWildcardsc(wrdARG, wrdLOW) ) {
#endif
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesc++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                } else {
                maskGLOBALlen = n;
                nameGLOBALlenc = wrdlen;
                if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_c(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesc++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                }
            }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower
```

```
// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                for (l=0; l < m-BB+1; l++) {
    WordsCheckedc++;
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTc[i][j] = LevenshteinTc[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTc[i][j] = min_AF(LevenshteinTc[i-1][j]+1, LevenshteinTc[i][j-1]+1, LevenshteinTc[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTc[i][j] = MIN(MIN((LevenshteinTc[i-1][j]+1),(LevenshteinTc[i][j-1]+1)),(LevenshteinTc[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTc[i][j] = MIN(MIN(LevenshteinTc[i-1][j],LevenshteinTc[i][j-1]),LevenshteinTc[i-1][j-1]); --LevenshteinTc[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTc[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTc[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTc[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesc++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVEc;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVEc:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsCheckedc++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTc[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDTc[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTc[i][j] = LevenshteinTc[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTc[i][j] = min_AF(LevenshteinTc[i-1][j]+1, LevenshteinTc[i][j-1]+1, LevenshteinTc[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTc[i][j] = MIN(MIN((LevenshteinTc[i-1][j]+1),(LevenshteinTc[i][j-1]+1)),(LevenshteinTc[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTc[i][j] = MIN(MIN(LevenshteinTc[i-1][j],LevenshteinTc[i][j-1]),LevenshteinTc[i-1][j-1]); --LevenshteinTc[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTc[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTc[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTc[BB][n]) {
```



```
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesc++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVEc;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDTc, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDTc, &wrdLOW[l], BB); wrdCACHEDTc[BB]=0;
// LD ]
}

                } // if (l < m-BB+1)
                }
                }
EXHAUSTIVEc:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsCheckedc++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTc[StartingPosition-1] && wrdCACHEDTc[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinTc[i][j] = LevenshteinTc[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTc[i][j] = min_AF(LevenshteinTc[i-1][j]+1, LevenshteinTc[i][j-1]+1, LevenshteinTc[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTc[i][j] = MIN(MIN((LevenshteinTc[i-1][j]+1),(LevenshteinTc[i][j-1]+1)),(LevenshteinTc[i-1][j-1]+1)); // Variant 2: This MS code is faster than above jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTc[i][j] = MIN(MIN(LevenshteinTc[i-1][j],LevenshteinTc[i][j-1]),LevenshteinTc[i-1][j-1]); --LevenshteinTc[i][j];} // Variant 3: This compound line is much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTc[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTc[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTc[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesc++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDTc, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
               }
```

```
                        else wrdlen++;
        } // k 'for'
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                        } // if (Exact_flag) {
       }

// Dth thread
#ifdef Commence_OpenMP
   #pragma omp section
#endif
       {
//This fragment is bugggy (somehow) [

                                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                                     k = WorkAreaLedgeTd;
                                                     while ( k < WorkAreaRedgeTd ) {
                                                     #ifdef RG7Gulliver
                                                                   FOUNDinPTR = Railgun_Trolldom_d(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTd-k+1, n); // 2019-May-17
                                                                   //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_d(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTd-k+1, n);
                                                                   //FOUNDinPTR = Railgun_Sekireigan_Wolfram_d(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTd-k+1, n); // 2019-May-17
                                                     #else
                                                                   FOUNDinPTR = Railgun_Quadruplet_7_d(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTd-k+1, n);
                                                     #endif
                                                     // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                     // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                     //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                                     if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                   i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                   while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                   while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                   k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                   //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                   if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                   memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLinesd++;
                                                                   j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                   j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                     // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP

                                                     #pragma omp critical

#endif
                                                     {
                                                                   if (Dump_flag)
                                                                   fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                   else
                                                                   fprintf( fp_outLINE, "%s\r\n", wrd);
                                                     }
}

                                                                   }
                                                     } else k = WorkAreaRedgeTd;
                                                     } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                        //{

       //memset (wrdCACHEDTd, 0, MaxLineLength+1+1);
       i = 1;
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       wrdlen = 0;
        for( k = WorkAreaLedgeTd; k <= WorkAreaRedgeTd; k++ )
       {
             workbyte = xgamsCACHE[k];

                        if( wrdlen < MAXboth) {
                                                     if (CaseSensitiveWildcardMatching_flag == 0)
                              wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                     else
                              wrdLOW[ wrdlen ] = ( workbyte );
                                        wrd[ wrdlen ] = workbyte;
                          }
                        if (workbyte == 10) {
                                        TotalLinesd++;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                          {
                                        wrd[ wrdlen ] = 0;
                                        wrdLOW[ wrdlen ] = 0;
                                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                     {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                        WordsCheckedd++;
                                        if (Exact_flag) {
                                                     //if ((long)( Railgun_Quadruplet_7Gulliver_d(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
```

```
                #ifdef RG7Gulliver
                    FOUNDinPTR = Railgun_Trolldom_d(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                    //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                    //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_d(wrdLOW, wrdARG, wrdlen, n);
                    //FOUNDinPTR = Railgun_Sekireigan_Wolfram_d(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                #else
                    FOUNDinPTR = Railgun_Quadruplet_7_d(wrdLOW, wrdARG, wrdlen, n);
                #endif
                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                //if ((long)( Railgun_Quadruplet_7_d(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                    {
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesd++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                    }
            } else {
                if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                if ( WildcardMatch_Iterative_Kazed(wrdARG, wrdLOW) ) {
#else
                if ( IterativeWildcardsd(wrdARG, wrdLOW) ) {
#endif
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesd++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                } else {
                maskGLOBALlen = n;
                nameGLOBALlend = wrdlen;
                if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_d(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesd++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                }
            }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
```

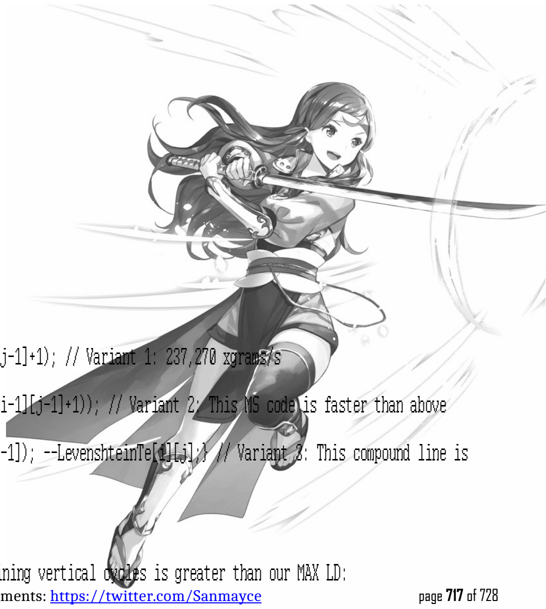
```
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                for (l=0; l < m-BB+1; l++) {
    WordsCheckedd++;
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTd[i][j] = LevenshteinTd[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTd[i][j] = min_AF(LevenshteinTd[i-1][j]+1, LevenshteinTd[i][j-1]+1, LevenshteinTd[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTd[i][j] = MIN(MIN((LevenshteinTd[i-1][j]+1),(LevenshteinTd[i][j-1]+1)),(LevenshteinTd[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTd[i][j] = MIN(MIN(LevenshteinTd[i-1][j],LevenshteinTd[i][j-1]),LevenshteinTd[i-1][j-1]); --LevenshteinTd[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTd[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTd[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTd[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesd++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVEd;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVEd:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsCheckedd++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTd[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDTd[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTd[i][j] = LevenshteinTd[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTd[i][j] = min_AF(LevenshteinTd[i-1][j]+1, LevenshteinTd[i][j-1]+1, LevenshteinTd[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTd[i][j] = MIN(MIN((LevenshteinTd[i-1][j]+1),(LevenshteinTd[i][j-1]+1)),(LevenshteinTd[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTd[i][j] = MIN(MIN(LevenshteinTd[i-1][j],LevenshteinTd[i][j-1]),LevenshteinTd[i-1][j-1]); --LevenshteinTd[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTd[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTd[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
```

```
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTd[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesd++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and 1 'for's:
                                goto EXHAUSTIVEd;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDTd, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDTd, &wrdLOW[1], BB); wrdCACHEDTd[BB]=0;
// LD ]
}

                } // if (1 < m-BB+1)
                }
                }
EXHAUSTIVEd:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsCheckedd++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTd[StartingPosition-1] && wrdCACHEDTd[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinTd[i][j] = LevenshteinTd[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTd[i][j] = min_AF(LevenshteinTd[i-1][j]+1, LevenshteinTd[i][j-1]+1, LevenshteinTd[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTd[i][j] = MIN(MIN((LevenshteinTd[i-1][j]+1),(LevenshteinTd[i][j-1]+1)),(LevenshteinTd[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTd[i][j] = MIN(MIN(LevenshteinTd[i-1][j],LevenshteinTd[i][j-1]),LevenshteinTd[i-1][j-1]); --LevenshteinTd[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTd[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTd[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTd[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesd++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDTd, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
```

```
                                        wrdlen = 0;
                        }
                        else wrdlen++;
        } // k 'for'
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                        } // if (Exact_flag) {
       }

// Eth thread
#ifdef Commence_OpenMP
    #pragma omp section
#endif
       {
//This fragment is bugggy (somehow) [

                                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                                      k = WorkAreaLedgeTe;
                                                      while ( k < WorkAreaRedgeTe ) {
                                                      #ifdef RG7Gulliver
                                                                   FOUNDinPTR = Railgun_Trolldom_e(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTe-k+1, n); // 2019-May-17
                                                                   //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_e(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTe-k+1, n);
                                                                   //FOUNDinPTR = Railgun_Sekireigan_Wolfram_e(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTe-k+1, n); // 2019-May-17
                                                      #else
                                                                   FOUNDinPTR = Railgun_Quadruplet_7_e(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTe-k+1, n);
                                                      #endif
                                                      // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                      // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                      //if ((long)( FOUNDinPTR - &xgamsCACHE[k] )>=0) {
                                                      if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                   i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                   while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                   while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                   k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                   //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                   if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                   memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLinese++;
                                                                   j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                   j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                      #pragma omp critical

#endif
                                                      {
                                                                   if (Dump_flag)
                                                                   fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                   else
                                                                   fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                                   }
                                                      } else k = WorkAreaRedgeTe;
                                                      } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                        //{

       //memset (wrdCACHEDTe, 0, MaxLineLength+1+1);
       i = 1;
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       wrdlen = 0;
        for( k = WorkAreaLedgeTe; k <= WorkAreaRedgeTe; k++ )
       {
             workbyte = xgamsCACHE[k];

                        if( wrdlen < MAXboth) {
                                                      if (CaseSensitiveWildcardMatching_flag == 0)
                               wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                      else
                               wrdLOW[ wrdlen ] = ( workbyte );
                                        wrd[ wrdlen ] = workbyte;
                           }
                        if (workbyte == 10) {
                                        TotalLinese++;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                           {
                                        wrd[ wrdlen ] = 0;
                                        wrdLOW[ wrdlen ] = 0;
                                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                      {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                        WordsCheckede++;
```

```
            if (Exact_flag) {
                //if ((long)( Railgun_Quadruplet_7Gulliver_e(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))>=0)
                #ifdef RG7Gulliver
                    FOUNDinPTR = Railgun_Trolldom_e(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                    //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                    //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_e(wrdLOW, wrdARG, wrdlen, n);
                    //FOUNDinPTR = Railgun_Sekireigan_Wolfram_e(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                #else
                    FOUNDinPTR = Railgun_Quadruplet_7_e(wrdLOW, wrdARG, wrdlen, n);
                #endif
                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                //if ((long)( FOUNDinPTR - wrdLOW ))>=0)
                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                //if ((long)( Railgun_Quadruplet_7_e(wrdLOW, wrdARG, wrdlen, n) - wrdLOW ))>=0)
                    {
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinese++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                    }
            } else {
                if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                if ( WildcardMatch_Iterative_Kazee(wrdARG, wrdLOW) ) {
#else
                if ( IterativeWildcardse(wrdARG, wrdLOW) ) {
#endif
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinese++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                } else {
                maskGLOBALlen = n;
                nameGLOBALlene = wrdlen;
                if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_e(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinese++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                }
            }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
```

```
// Edelweiß !!
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                for (l=0; l < m-BB+1; l++) {
    WordsCheckede++;
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTe[i][j] = LevenshteinTe[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTe[i][j] = min_AF(LevenshteinTe[i-1][j]+1, LevenshteinTe[i][j-1]+1, LevenshteinTe[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTe[i][j] = MIN(MIN((LevenshteinTe[i-1][j]+1),(LevenshteinTe[i][j-1]+1)),(LevenshteinTe[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTe[i][j] = MIN(MIN(LevenshteinTe[i-1][j],LevenshteinTe[i][j-1]),LevenshteinTe[i-1][j-1]); --LevenshteinTe[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTe[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTe[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTe[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinese++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVEe;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVEe:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsCheckede++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTe[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDTe[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTe[i][j] = LevenshteinTe[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTe[i][j] = min_AF(LevenshteinTe[i-1][j]+1, LevenshteinTe[i][j-1]+1, LevenshteinTe[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTe[i][j] = MIN(MIN((LevenshteinTe[i-1][j]+1),(LevenshteinTe[i][j-1]+1)),(LevenshteinTe[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTe[i][j] = MIN(MIN(LevenshteinTe[i-1][j],LevenshteinTe[i][j-1]),LevenshteinTe[i-1][j-1]); --LevenshteinTe[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
```



```
        if ( LevenshteinTe[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTe[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTe[BB][n]) {
                                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinese++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                                #pragma omp critical
#endif
                                                {
                                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                                }
}
                                                // Once dumping the line we need double 'break' from BB and 1 'for's:
                                                goto EXHAUSTIVEe;
                                                //if ((DumpedLines & 0xff) == 0xff)
                                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDTe, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDTe, &wrdLOW[1], BB); wrdCACHEDTe[BB]=0;
// LD ]
}

                        } // if (1 < m-BB+1)
                        }
                        }
EXHAUSTIVEe:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsCheckede++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTe[StartingPosition-1] && wrdCACHEDTe[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinTe[i][j] = LevenshteinTe[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTe[i][j] = min_AF(LevenshteinTe[i-1][j]+1, LevenshteinTe[i][j-1]+1, LevenshteinTe[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTe[i][j] = MIN(MIN((LevenshteinTe[i-1][j]+1),(LevenshteinTe[i][j-1]+1)),(LevenshteinTe[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTe[i][j] = MIN(MIN(LevenshteinTe[i-1][j],LevenshteinTe[i][j-1]),LevenshteinTe[i-1][j-1]); --LevenshteinTe[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTe[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTe[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTe[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinese++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //          //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //          fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDTe, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
```

```
                           }
// Wildcard search ]
                                        wrdlen = 0;
                        }
                        else wrdlen++;
        } // k 'for'
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                        } // if (Exact_flag) {
       }

// Fth thread
#ifdef Commence_OpenMP
    #pragma omp section
#endif
       {
//This fragment is bugggy (somehow) [

                                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[ Since r.1-++
                                                      k = WorkAreaLedgeTf;
                                                      while ( k < WorkAreaRedgeTf ) {
                                                      #ifdef RG7Gulliver
                                                                   FOUNDinPTR = Railgun_Trolldom_f(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTf-k+1, n); // 2019-May-17
                                                                   //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_f(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTf-k+1, n);
                                                                   //FOUNDinPTR = Railgun_Sekireigan_Wolfram_f(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTf-k+1, n); // 2019-May-17
                                                      #else
                                                                   FOUNDinPTR = Railgun_Quadruplet_7_f(&xgamsCACHE[k], wrdARG, WorkAreaRedgeTf-k+1, n);
                                                      #endif
                                                      // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                      // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                      //if ((long)( FOUNDinPTR - &xgamsCACHE[k] ))=0) {
                                                      if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                                                   i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                                                   while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                                                   while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                                                   k = j+1; // Should "point" to first symbol after the dumped fragment.

                                                                   //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                                                   if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                                                   memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLinesf++;
                                                                   j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                                                   j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
                                                      // Below pragma is needed explicitly only for MinGW, grrr...
if (YesDump == 1) {
#ifdef Commence_OpenMP

                                                      #pragma omp critical

#endif
                                                      {
                                                                   if (Dump_flag)
                                                                   fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                                                   else
                                                                   fprintf( fp_outLINE, "%s\r\n", wrd);
                                                      }
}
                                                                   }
                                                      } else k = WorkAreaRedgeTf;
                                                      } // while
// WHOLE buffer at once not line-by-line ]]]]]]] Since r.1-++
                                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                                        //{

       //memset (wrdCACHEDTf, 0, MaxLineLength+1+1);
       i = 1;
        //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       wrdlen = 0;
        for( k = WorkAreaLedgeTf; k <= WorkAreaRedgeTf; k++ )
       {
             workbyte = xgamsCACHE[k];

                        if( wrdlen < MAXboth) {
                                                      if (CaseSensitiveWildcardMatching_flag == 0)
                               wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                      else
                               wrdLOW[ wrdlen ] = ( workbyte );
                                        wrd[ wrdlen ] = workbyte;
                           }
                        if (workbyte == 10) {
                                        TotalLinesf++;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                           {
                                        wrd[ wrdlen ] = 0;
                                        wrdLOW[ wrdlen ] = 0;
                                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                      {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
```

```
// WILDCARD IP [
            WordsCheckedf++;
            if (Exact_flag) {
                //if ((long)( Railgun_Quadruplet_7Gulliver_f(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                #ifdef RG7Gulliver
                    FOUNDinPTR = Railgun_Trolldom_f(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                    //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                    //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_f(wrdLOW, wrdARG, wrdlen, n);
                    //FOUNDinPTR = Railgun_Sekireigan_Wolfram_f(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                #else
                    FOUNDinPTR = Railgun_Quadruplet_7_f(wrdLOW, wrdARG, wrdlen, n);
                #endif
                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                //if ((long)( Railgun_Quadruplet_7_f(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                    {
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesf++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                    }
            } else {
                if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                if ( WildcardMatch_Iterative_Kazef(wrdARG, wrdLOW) ) {
#else
                if ( IterativeWildcardsf(wrdARG, wrdLOW) ) {
#endif
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesf++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                } else {
                maskGLOBALlen = n;
                nameGLOBALlenf = wrdlen;
                if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_f(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesf++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                #pragma omp critical
#endif
                {
                    if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
                }
}
                }
                }
            }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don’t enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we’ll walk through the whole length of ’m’, ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here ’m’).
// We need to factorize ’m’ down to all ’n+AtMostLevenshteinDistance’ long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// ¦Edelweiß ¦¦ [[edelweiss]] ¦¦ edelweiss flower
// Let’s search fuzzily for "edelvais", pretending we don’t know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// ’edelvais’ vs ’edelweiss’:
// - the mispelled one has one character less;
// - the mispelled one has 2 wrong characters: ’v’ & ’a’ instead of ’w’ & ’e’.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, ’m - Order + 1’ is the number of total BBs for text ’m’ bytes long: 46-11+1 = 36
```

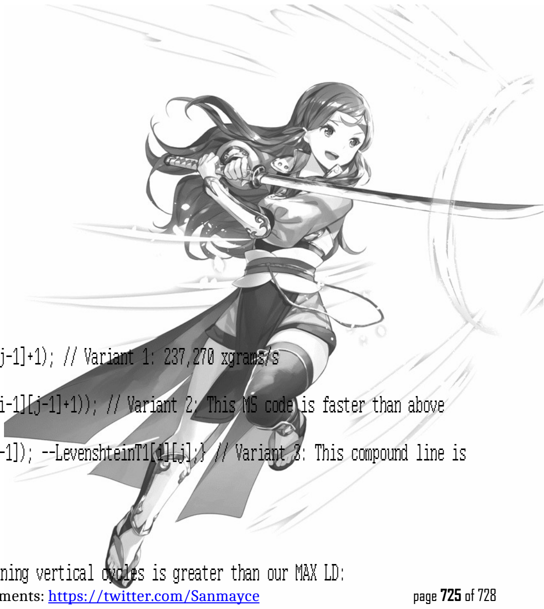
```
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                for (l=0; l < m-BB+1; l++) {
    WordsCheckedf++;
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTf[i][j] = LevenshteinTf[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTf[i][j] = min_AF(LevenshteinTf[i-1][j]+1, LevenshteinTf[i][j-1]+1, LevenshteinTf[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTf[i][j] = MIN(MIN((LevenshteinTf[i-1][j]+1),(LevenshteinTf[i][j-1]+1)),(LevenshteinTf[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTf[i][j] = MIN(MIN(LevenshteinTf[i-1][j],LevenshteinTf[i][j-1]),LevenshteinTf[i-1][j-1]); --LevenshteinTf[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTf[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTf[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTf[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesf++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                goto EXHAUSTIVEf;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVEf:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsCheckedf++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTf[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDTf[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinTf[i][j] = LevenshteinTf[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTf[i][j] = min_AF(LevenshteinTf[i-1][j]+1, LevenshteinTf[i][j-1]+1, LevenshteinTf[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTf[i][j] = MIN(MIN((LevenshteinTf[i-1][j]+1),(LevenshteinTf[i][j-1]+1)),(LevenshteinTf[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTf[i][j] = MIN(MIN(LevenshteinTf[i-1][j],LevenshteinTf[i][j-1]),LevenshteinTf[i-1][j-1]); --LevenshteinTf[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }
```

```
        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTf[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTf[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTf[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesf++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVEf;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDTf, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDTf, &wrdLOW[1], BB); wrdCACHEDTf[BB]=0;
// LD ]
}

                } // if (l < m-BB+1)
                }
                }
EXHAUSTIVEf:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsCheckedf++;
// LD [
StartingPosition = 1;
while (wrdCACHEDTf[StartingPosition-1] && wrdCACHEDTf[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinTf[i][j] = LevenshteinTf[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinTf[i][j] = min_AF(LevenshteinTf[i-1][j]+1, LevenshteinTf[i][j-1]+1, LevenshteinTf[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinTf[i][j] = MIN(MIN((LevenshteinTf[i-1][j]+1),(LevenshteinTf[i][j-1]+1)),(LevenshteinTf[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinTf[i][j] = MIN(MIN(LevenshteinTf[i-1][j],LevenshteinTf[i][j-1]),LevenshteinTf[i-1][j-1]); --LevenshteinTf[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinTf[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinTf[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinTf[m][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
DumpedLinesf++;
if (YesDump == 1) {
#ifdef Commence_OpenMP
                                #pragma omp critical
#endif
                                {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
                                }
}
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDTf, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
```

```
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
              }
              else wrdlen++;
     } // k 'for'
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        } // if (Exact_flag) {
    }

  }
}// pragma

    memcpy(xgamsCACHE, &xgamsCACHE[ (CACHEsize-1-CACHEremainder)+1 ], CACHEremainder);
} //while (size_inLINESIXFOURleftforparsing >= CACHEsize-CACHEremainder) {

// We have not finished the 'size_inLINESIXFOURleftforparsing' loop yet! Must check for remainder! [
if (size_inLINESIXFOURleftforparsing != 0) {
    fread( xgamsCACHE+CACHEremainder, 1, size_inLINESIXFOURleftforparsing, fp_inLINE );

    // TO-DO: Use the vectorized LOWERcasing that I have [
    if ( CaseSensitiveWildcardMatching_flag==0 ) {

        for(i=0;i<size_inLINESIXFOURleftforparsing;i++) {
                xgamsCACHE[ i ] = KAZE_toupper( xgamsCACHE[ i ] );
        }

        //UCASE_XMM_inplace(xgamsCACHE, size_inLINESIXFOURleftforparsing); // It is not padded, therefore fallback to scalar.
    }
    // TO-DO: Use the vectorized LOWERcasing that I have ]

    // Working area: xgamsCACHE..xgamsCACHE +(CACHEremainder+size_inLINESIXFOURleftforparsing) -1
    //fwrite( xgamsCACHE, 1, xgamsCACHE +(CACHEremainder+size_inLINESIXFOURleftforparsing) -1 - xgamsCACHE +1, fp_outLINE ); //DELDEL
    //...

//This fragment is bugggy (somehow) [

                        if (Exact_flag) {
// WHOLE buffer at once not line-by-line [[[[[[[[ Since r.1-++
                                k = 0;
                                while ( k < (CACHEremainder+size_inLINESIXFOURleftforparsing) -1 ) {
                                #ifdef RG7Gulliver
                                        FOUNDinPTR = Railgun_Trolldom_1(&xgamsCACHE[k], wrdARG, (CACHEremainder+size_inLINESIXFOURleftforparsing) -1-k+1, n); // 2019-May-17
                                        //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_1(&xgamsCACHE[k], wrdARG, (CACHEremainder+size_inLINESIXFOURleftforparsing) -1-k+1, n);
                                        //FOUNDinPTR = Railgun_Sekireigan_Wolfram_1(&xgamsCACHE[k], wrdARG, (CACHEremainder+size_inLINESIXFOURleftforparsing) -1-k+1, n); // 2019-
    May-17
                                #else
                                        FOUNDinPTR = Railgun_Quadruplet_7_1(&xgamsCACHE[k], wrdARG, (CACHEremainder+size_inLINESIXFOURleftforparsing) -1-k+1, n);
                                #endif
                                // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                //if ((long)( FOUNDinPTR - &xgamsCACHE[k] ))=0) {
                                if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[k])) {
                                        i = k + (long)( FOUNDinPTR - &xgamsCACHE[k] ); j = i;
                                        while (xgamsCACHE[j] != 10) {++j;} // Works both on UNIX(LF) and Windows(CRLF)
                                        while (i > k && xgamsCACHE[i-1] != 10) {--i;}
                                        k = j+1; // Should "point" to first symbol after the dumped fragment.

                                        //fwrite( &xgamsCACHE[i], j - i + 1, 1, fp_outLINE ); DumpedLines1++;
                                        if (j - i + 1 <= 168*MaxLineLength) { // fix for 1-++
                                        memcpy(wrd, &xgamsCACHE[i], j - i + 1); wrd[j - i + 1]=0; DumpedLines++;
                                        j--; if (wrd[j - i + 1]==10) wrd[j - i + 1]=0;
                                        j--; if (wrd[j - i + 1]==13) wrd[j - i + 1]=0;
if (YesDump == 1) {
                                        if (Dump_flag)
                                        fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]);
                                        else
                                        fprintf( fp_outLINE, "%s\r\n", wrd);
}
                                        }
                                } else k = (CACHEremainder+size_inLINESIXFOURleftforparsing) -1;
                                } // while
// WHOLE buffer at once not line-by-line ]]]]]]]] Since r.1-++
                        } else { // if (Exact_flag) {

//This fragment is bugggy (somehow) ]
                        //{

    //memset (wrdCACHEDT1, 0, MaxLineLength+1+1);
    i = 1;
     //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    wrdlen = 0;
    StartOfLineFLAG=1;
     for( k = 0; k <= (CACHEremainder+size_inLINESIXFOURleftforparsing) -1; k++ )
    {
        if (StartOfLineFLAG == 1) { kStartOfLine = k; StartOfLineFLAG=0; }
        workbyte = xgamsCACHE[k];
```

```
                        if( wrdlen < MAXboth) {
                                                    if (CaseSensitiveWildcardMatching_flag == 0)
                              wrdLOW[ wrdlen ] = KAZE_toupper( workbyte );
                                                    else
                              wrdLOW[ wrdlen ] = ( workbyte );
                                        wrd[ wrdlen ] = workbyte;
                          }
                        if (workbyte == 10) {
                                        TotalLines++;
                                        StartOfLineFLAG=1;
// Wildcard search [
                        if ( 0 < wrdlen && wrdlen < MAXboth)
                          {
                                        wrd[ wrdlen ] = 0;
                                        wrdLOW[ wrdlen ] = 0;
                                        if ( wrd[ wrdlen-1 ] == 13 ) //CR
                                                    {--wrdlen; wrd[ wrdlen ] = 0; wrdLOW[ wrdlen ] = 0;}

if (WILDCARD_IP_flag) {
// WILDCARD IP [
                                        WordsChecked++;
                                        if (Exact_flag) {
                                                    //if ((long)( Railgun_Quadruplet_7Gulliver_1(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                    #ifdef RG7Gulliver
                                                                FOUNDinPTR = Railgun_Trolldom(&xgamsCACHE[kStartOfLine], wrdARG, wrdlen, n); // 2019-May-17
                                                                //FOUNDinPTR = Railgun_Doublet(&xgamsCACHE[kStartOfLine], wrdARG, wrdlen, n); // 2019-May-17
                                                                //FOUNDinPTR = strstr(wrdLOW, wrdARG);
                                                                //FOUNDinPTR = Railgun_Quadruplet_7Gulliver_1(wrdLOW, wrdARG, wrdlen, n);
                                                                //FOUNDinPTR = Railgun_Sekireigan_Wolfram_1(wrdLOW, wrdARG, wrdlen, n); // 2019-May-17
                                                    #else
                                                                FOUNDinPTR = Railgun_Quadruplet_7_1(wrdLOW, wrdARG, wrdlen, n);
                                                    #endif
                                                    // Commented line below works under MinGW & Intel 12.1 for Windows but fails under Linux:
                                                    // Linux thinks that 0 - ~3,000,000,000 = ~1,000,000,000
                                                    //if ((long)( FOUNDinPTR - wrdLOW )>=0)
                                                    if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(&xgamsCACHE[kStartOfLine]))
                                                    //if ((unsigned long long)( FOUNDinPTR) >=(unsigned long long)(wrdLOW))
                                                    //if ((long)( Railgun_Quadruplet_7_1(wrdLOW, wrdARG, wrdlen, n) - wrdLOW )>=0)
                                                                {DumpedLines++;
if (YesDump == 1) {
if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
}
                                                                }
                                        } else {
                                                    if (WILDCARD_FAST_flag) {
#if defined(_WildFastKaze_)
                                                    if ( WildcardMatch_Iterative_Kaze1(wrdARG, wrdLOW) ) {
#else
                                                    if ( IterativeWildcards1(wrdARG, wrdLOW) ) {
#endif
                                                                DumpedLines++;
if (YesDump == 1) {
if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
}
                                                    }
                                                    } else {
                                                    maskGLOBALlen = n;
                                                    nameGLOBALlen1 = wrdlen;
                                                    if ( EnhancedMaskTest_OrEmpty_AndNotEmpty_1(wrdARG, 0, wrdLOW, 0) ) { // Caution: Not lowercased as it should!
                                                                DumpedLines++;
if (YesDump == 1) {
if (Dump_flag) fprintf( fp_outLINE, "[%s] %s /%s/\r\n", wrdARG, wrd, argv[3-WILDCARD_IP_flag]); else fprintf( fp_outLINE, "%s\r\n", wrd);
}
                                                    }
                                                    }
                                        }
// WILDCARD IP ]
} else {

// A simple heuristic #1: Don't enter the nasty loops unless MaximumLevenshteinDistance >= ABS(m-n).
m = wrdlen; // strlen(wrd);
//if (m>MaxLineLength)
//{ printf( "\nKazahana: Incoming xgram exceeding the limit.\n" ); exit( 2 ); }
// Above two commented lines are too severe, changed with next line allowing to search into lines bigger than our needs:
if (m) {
if (EXHAUSTIVE_flag == 1) {

// Here we'll walk through the whole length of 'm', ay-yaa.
// EXHAUSTIVE [[[[[[[[[[[[[[[[[[[[[[[[[[[[[
// Here the old fuzzy is replaced with exhaustive one, using BBs of the incoming string (here 'm').
// We need to factorize 'm' down to all 'n+AtMostLevenshteinDistance' long strings/BBs and to search into them ONE-BY-ONE - a gruelling task indeed!
//// Example:
// One of those 33/5 lines is a 46 bytes long string, i.e. m = 46:
// |Edelweiß || [[edelweiss]] || edelweiss flower
// Let's search fuzzily for "edelvais", pretending we don't know the right spelling:
// To find "edelweiss" we need at least Levenshtein distance 3, i.e. n = 8, AtMostLevenshteinDistance = 3:
// 'edelvais' vs 'edelweiss':
// - the mispelled one has one character less;
```

```
// - the mispelled one has 2 wrong characters: 'v' & 'a' instead of 'w' & 'e'.
// From above we need Building-Blocks of 46 bytes order 8+3.
// Inhere we are using order 11, 'm - Order + 1' is the number of total BBs for text 'm' bytes long: 46-11+1 = 36
// The 36 BBs:
// ¦Edelweiß ¦
// Edelweiß ¦¦
// ...
// weiss flowe
// eiss flower

// Slow! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/*
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                for (l=0; l < m-BB+1; l++) {
    WordsChecked++;
// LD [
    SkipHeuristic=0;
    for(i=1;i<=BB;i++) {
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT1[i][j] = LevenshteinT1[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT1[i][j] = min_AF(LevenshteinT1[i-1][j]+1, LevenshteinT1[i][j-1]+1, LevenshteinT1[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT1[i][j] = MIN(MIN((LevenshteinT1[i-1][j]+1),(LevenshteinT1[i][j-1]+1)),(LevenshteinT1[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT1[i][j] = MIN(MIN(LevenshteinT1[i-1][j],LevenshteinT1[i][j-1]),LevenshteinT1[i-1][j-1]); --LevenshteinT1[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }
        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. BB) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT1[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT1[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT1[BB][n]) {
                                {
DumpedLines++;
if (YesDump == 1) {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
}
                                }
                                goto EXHAUSTIVE;
                                //break; // No need of further checking down the line, one dump only is needed.
                                //if ((DumpedLines & 0xff) == 0xff)
                                //        //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //        fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
// LD ]
                }
                }
EXHAUSTIVE:;
*/
// Slow! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// Fast! [<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

                for (l=0; l < m-(n-AtMostLevenshteinDistance)+1; l++) { // Here BB = n-AtMostLevenshteinDistance the smallest BB
                for (BB=n-AtMostLevenshteinDistance; BB <=n+AtMostLevenshteinDistance; BB++) {
                // From here on 'm' will be replaced by BB
                if (l < m-BB+1) {

if (AtMostLevenshteinDistance >= MAX(BB,n)-MIN(BB,n)) // This is the only add-on for r.1+
{
    WordsChecked++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT1[StartingPosition-1] && wrdLOW[l+StartingPosition-1] && wrdCACHEDT1[StartingPosition-1]==wrdLOW[l+StartingPosition-1] )
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=BB;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[l+i-1] == wrdARG[j-1])
                        LevenshteinT1[i][j] = LevenshteinT1[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT1[i][j] = min_AF(LevenshteinT1[i-1][j]+1, LevenshteinT1[i][j-1]+1, LevenshteinT1[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT1[i][j] = MIN(MIN((LevenshteinT1[i-1][j]+1),(LevenshteinT1[i][j-1]+1)),(LevenshteinT1[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT1[i][j] = MIN(MIN(LevenshteinT1[i-1][j],LevenshteinT1[i][j-1]),LevenshteinT1[i-1][j-1]); --LevenshteinT1[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
```



```
        if ( LevenshteinT1[i][n] - (BB-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT1[i][n] - (BB-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT1[BB][n]) {
                                // Below pragma is needed explicitly only for MinGW, grrr...
//#ifdef Commence_OpenMP
//                              #pragma omp critical
//#endif
//                              {
DumpedLines++;
if (YesDump == 1) {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
}
//                              }
                                // Once dumping the line we need double 'break' from BB and l 'for's:
                                goto EXHAUSTIVE;
                                //if ((DumpedLines & 0xff) == 0xff)
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    //memcpy(wrdCACHEDT1, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
    memcpy(wrdCACHEDT1, &wrdLOW[l], BB); wrdCACHEDT1[BB]=0;
// LD ]
}

                } // if (l < m-BB+1)
                }
                }
EXHAUSTIVE:;

// Fast! ]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

// EXHAUSTIVE ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

} else {
if (m<=MaxLineLength)
if (AtMostLevenshteinDistance >= MAX(m,n)-MIN(m,n)) // This is the only add-on for r.1+
{
    WordsChecked++;
// LD [
StartingPosition = 1;
while (wrdCACHEDT1[StartingPosition-1] && wrdCACHEDT1[StartingPosition-1]==wrdLOW[StartingPosition-1] ) // No need of && wrd[StartingPosition-1]
    StartingPosition++;
// The bail out 'i' value (heuristic #2) affects our cached value here, 'StartingPosition' cannot be greater than 'i':
StartingPosition = MIN(StartingPosition, i);

    SkipHeuristic=0;
    for(i=StartingPosition;i<=m;i++) { // StartingPosition is in range 1..
        for (j=1;j<=n;j++) {
                if(wrdLOW[i-1] == wrdARG[j-1])
                        LevenshteinT1[i][j] = LevenshteinT1[i-1][j-1];
                else
#if defined(_WIN32ASM_)
                        LevenshteinT1[i][j] = min_AF(LevenshteinT1[i-1][j]+1, LevenshteinT1[i][j-1]+1, LevenshteinT1[i-1][j-1]+1); // Variant 1: 237,270 xgrams/s
#else
                        LevenshteinT1[i][j] = MIN(MIN((LevenshteinT1[i-1][j]+1),(LevenshteinT1[i][j-1]+1)),(LevenshteinT1[i-1][j-1]+1)); // Variant 2: This MS code is faster than above
    jumpless code! // 358,327 xgrams/s
                        //{LevenshteinT1[i][j] = MIN(MIN(LevenshteinT1[i-1][j],LevenshteinT1[i][j-1]),LevenshteinT1[i-1][j-1]); --LevenshteinT1[i][j];} // Variant 3: This compound line is
    much slower than above inc-inc-inc code! // 237,270 xgrams/s
#endif
        }

        // A simple heuristic #2: Discontinue the nasty vertical loop (i.e. m) when the LD in cell in the last column minus the remaining vertical cycles is greater than our MAX LD:
        if ( LevenshteinT1[i][n] - (m-i) >= 0 && AtMostLevenshteinDistance < LevenshteinT1[i][n] - (m-i) ) {SkipHeuristic=1; break;} // Caution: Levenshtein[i][n] can be less than (m-i), this changes
    nothing the logic is the same.
    }
                        if (SkipHeuristic==0 && AtMostLevenshteinDistance >= LevenshteinT1[m][n]) {
                                {
DumpedLines++;
if (YesDump == 1) {
                                fprintf( fp_outLINE, "%s\r\n", wrd);
}
                                }
                                //if ((DumpedLines & 0xff) == 0xff)
                                //      //printf( "Dumped lines i.e. hits so far: %s\r", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
                                //      fflush(fp_outLINE); // Not sure: CTRL+C doesn't flush?!
                        }
    // The bail out 'i' value (heuristic #2) affects our cached value here, 'i' is the needed one:
    memcpy(wrdCACHEDT1, wrdLOW, m+1); // +1 because we need the ASCII 000 termination;
// LD ]
}
} //if (EXHAUSTIVE_flag == 1)
}
} //if (WILDCARD_IP_flag)
                }
// Wildcard search ]
                        wrdlen = 0;
               }
```

```
                        else wrdlen++;
        } // k 'for'
        //^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                                        } // if (Exact_flag) {

}
       //Melnitchka = Melnitchka & 3; // 0 1 2 3: 00 01 10 11
       //printf( "%s; Speed: %s bytes/clock; Traversed: %s bytes\r", Auberge[Melnitchka++], _ui64toaKAZEzerocomma(((size_inLINESIXFOUR-0)>>0)/((long)(clocks3 - clocks1 + 1)), llTOaDigits, 10) +12,
       _ui64toaKAZEcomma((size_inLINESIXFOUR-0), llTOaDigits2, 10)); //2019-May-17
// We have not finished the 'size_inLINESIXFOURleftforparsing' loop yet! Must check for remainder! ]
// MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING MT PARSING ]

       fclose(fp_inLINE);
       fclose(fp_outLINE);
       //free(xgamsCACHE); // Cannot free it like that because it is not the same as when malloc() was used!
       (void) time(&t3);
       if (t3 <= t1) {t3 = t1; t3++;}
       clocks2 = clock();
       //printf( "\n" );

       DumpedLines=DumpedLines+DumpedLines1+DumpedLines2+DumpedLines3+DumpedLines4+DumpedLines5+DumpedLines6+DumpedLines7+DumpedLines8+DumpedLines9+DumpedLines0+DumpedLinesa+DumpedLinesb+DumpedLinesc+Dumped
       Linesd+DumpedLinese+DumpedLinesf;

       TotalLines=TotalLines+TotalLines1+TotalLines2+TotalLines3+TotalLines4+TotalLines5+TotalLines6+TotalLines7+TotalLines8+TotalLines9+TotalLines0+TotalLinesa+TotalLinesb+TotalLinesc+TotalLinesd+TotalLine
       se+TotalLinesf;

       WordsChecked=WordsChecked+WordsChecked1+WordsChecked2+WordsChecked3+WordsChecked4+WordsChecked5+WordsChecked6+WordsChecked7+WordsChecked8+WordsChecked9+WordsChecked0+WordsCheckeda+WordsCheckedb+Words
       Checkedc+WordsCheckedd+WordsCheckede+WordsCheckedf;

       Melnitchka = Melnitchka & 3; // 0 1 2 3: 00 01 10 11
       //printf( "%s; Speed: %s bytes/clock; Traversed: %s bytes; Dumped: %s\n", Auberge[Melnitchka++], _ui64toaKAZEzerocomma(((size_inLINESIXFOUR-0)>>0)/((long)(clocks3 - clocks1 + 1)), llTOaDigits, 10)
       +12, _ui64toaKAZEcomma((size_inLINESIXFOUR-0), llTOaDigits2, 10)); //2019-May-17
       printf( "%s; Speed: %s bytes/second; Traversed: %s bytes; Dumped: %s\n", Auberge[Melnitchka++], _ui64toaKAZEzerocomma(((size_inLINESIXFOUR-size_inLINESIXFOURleftforparsing)>>0)*CLOCKS_PER_SEC/((long)
       (clocks3 - clocks1 + 1)), llTOaDigits, 10) +12, _ui64toaKAZEcomma((size_inLINESIXFOUR-size_inLINESIXFOURleftforparsing), llTOaDigits2, 10), _ui64toaKAZEcomma(DumpedLines, llTOaDigits3, 10));
       fflush(stdout);

/*
                                        if (Exact_flag)
{
if (YesDump == 1)
            printf( "Kazahana: Dumped xgrams: %s\n", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
else
            printf( "Kazahana: Counted xgrams: %s\n", _ui64toaKAZEcomma(DumpedLines, llTOaDigits, 10) );
}
                                        else
*/
            printf( "Kazahana: Total/Checked/Dumped xgrams: %s/%s/%s\n", _ui64toaKAZEcomma(TotalLines, llTOaDigits3, 10), _ui64toaKAZEcomma(WordsChecked, llTOaDigits2, 10), _ui64toaKAZEcomma(DumpedLines,
       llTOaDigits, 10) );
            printf( "Kazahana: Performance: %s KB/s\n", _ui64toaKAZEcomma((size_inLINESIXFOUR>>10)*CLOCKS_PER_SEC/((long)(clocks2 - clocks1 + 1)), llTOaDigits, 10) );
/*
                                        if (Exact_flag) {} else
*/
            printf( "Kazahana: Performance: %s xgrams/s\n", _ui64toaKAZEcomma((TotalLines)*CLOCKS_PER_SEC/((long)(clocks2 - clocks1 + 1)), llTOaDigits, 10) ); // CLOCKS_PER_SEC
            printf( "Kazahana: Performance: Total/fread() clocks: %s/%s\n", _ui64toaKAZEcomma((long)(clocks2 - clocks1 + 1), llTOaDigits, 10), _ui64toaKAZEcomma((long)FREADclocks, llTOaDigits2, 10) );
            printf( "Kazahana: Performance: I/O time, i.e. fread() time, is %s percents\n", _ui64toaKAZEcomma(FREADclocks*100/(long)(clocks2 - clocks1 + 1), llTOaDigits, 10) );
#if defined(_icl_mumbo_jumbo_)
            printf( "Kazahana: Performance: RDTSC I/O time, i.e. fread() time, is %s ticks\n", _ui64toaKAZEcomma(ticksTOTAL, llTOaDigits, 10) );
#endif
            printf( "Kazahana: Done.\n" );
       }
       exit (0);
}


// Test on laptop with Q9550s 2833MHz, 4/4 cores/threads, Windows 7 64bit:
/*
D:\_KAZE\GameraWikipediaWiktionary>type Kazahana_2014-Dec-04\Kazahana_compile_GCC.bat
gcc -O3 -funroll-loops -static -o Kazahana_Hexadecad_GCC_472 Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE.c -fopenmp -DCommence_OpenMP -D_FILE_OFFSET_BITS=64 -D_gcc_mumbo_jumbo_
gcc -O3 -funroll-loops -static -o Kazahana_Monad_GCC_472 Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE.c -fopenmp -D_FILE_OFFSET_BITS=64 -D_gcc_mumbo_jumbo_

D:\_KAZE\GameraWikipediaWiktionary>type Kazahana_2014-Dec-04\Kazahana_compile_Intel12_64bit.bat
icl /O3 /arch:SSE2 /QxSSE2 /Qunroll /MT Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE.c /FAcs /FeKazahana_r1-+
       +fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_HEXADECAD-Threads_IntelV12_SSE2_64bit /Qopenmp /Qopenmp-link:static -DCommence_OpenMP -D_icl_mumbo_jumbo_
icl /O3 /arch:SSE2 /QxSSE2 /Qunroll /MT Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE.c /FeKazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_MONAD-
       Thread_IntelV12_SSE2_64bit -D_icl_mumbo_jumbo_

D:\_KAZE\GameraWikipediaWiktionary>timer32.exe Kazahana_Hexadecad_GCC_472.exe 4e "Silvestor Staloune" enwiki-20141008-pages-articles.xml 11263
Kazahana, a superfast exact & wildcards & Levenshtein Distance (Wagner-Fischer) searcher, r. 1-++fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix, copyleft Kaze 2014-Nov-19.
Pattern: Silvestor Staloune
omp_get_num_procs( ) = 4
omp_get_max_threads( ) = 4
Enforcing HEXADECAD i.e. hexadecuple-threads ...
Allocating Master-Buffer 11263KB ... OK
\; 00,000,001,376 bytes/clock
Kazahana: Total/Checked/Dumped xgrams: 800,855,553/342,059,464,575/2,106
Kazahana: Performance: 1 KB/clock
Kazahana: Performance: 21 xgrams/clock
Kazahana: Performance: Total/fread() clocks: 36,459,222/1,379,563
Kazahana: Performance: I/O time, i.e. fread() time, is 3 percents
```

```
Kazahana: Done.

Kernel  Time =    38.345 =    0%
User    Time =136250.493 =  373%
Process Time =136288.838 =  373%    Virtual  Memory =     14 MB
Global  Time = 36460.185 =  100%    Physical Memory =     16 MB

D:\_KAZE\GameraWikipediaWiktionary>dir Kazahana.txt
 Volume in drive D is S640_Vol5
 Volume Serial Number is 5861-9E6C

 Directory of D:\_KAZE\GameraWikipediaWiktionary

12/03/2014  01:10 PM         1,064,420 Kazahana.txt
               1 File(s)      1,064,420 bytes
               0 Dir(s)  63,694,749,696 bytes free

D:\_KAZE\GameraWikipediaWiktionary>timer32.exe Kazahana_r1-++fix+nowait_critical_nixFIX_WolfRAM+fixITER+EX+CS_fix_DEFINE_HEXADECAD-Threads_IntelV12_SSE2_64bit 4e "Silvestor Staloune" enwiki-20141008-pages-articles.xml 11263
Kazahana, a superfast exact & wildcards & Levenshtein Distance (Wagner-Fischer) searcher, r. 1-++fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix_DEFINE, copyleft Kaze 2014-Dec-03.
Pattern: Silvestor Staloune
omp_get_num_procs( ) = 4
omp_get_max_threads( ) = 4
Enforcing HEXADECAD i.e. hexadecuple-threads ...
Allocating Master-Buffer 11263KB ... OK
\; Speed: 00,000,002,001 bytes/clock; Traversed: 50,144,448,379 bytes
Kazahana: Total/Checked/Dumped xgrams: 800,855,553/342,059,464,575/2,106
Kazahana: Performance: 1 KB/clock
Kazahana: Performance: 31 xgrams/clock
Kazahana: Performance: Total/fread() clocks: 25,073,428/602,292
Kazahana: Performance: I/O time, i.e. fread() time, is 2 percents
Kazahana: Performance: RDTSC I/O time, i.e. fread() time, is 1,704,219,997,078 ticks
Kazahana: Done.

Kernel  Time =   284.670 =    1%
User    Time = 92233.204 =  367%
Process Time = 92517.875 =  368%    Virtual  Memory =     17 MB
Global  Time = 25073.682 =  100%    Physical Memory =     16 MB

D:\_KAZE\GameraWikipediaWiktionary>dir Kazahana.txt
 Volume in drive D is S640_Vol5
 Volume Serial Number is 5861-9E6C

 Directory of D:\_KAZE\GameraWikipediaWiktionary

12/04/2014  08:51 AM         1,064,420 Kazahana.txt
               1 File(s)      1,064,420 bytes
               0 Dir(s)  67,609,645,056 bytes free

D:\_KAZE\GameraWikipediaWiktionary>
*/
```