



KAZUYA (POWERED BY QUICKLZ)

A 32/64 BIT COMPRESSED SENTENCE DUMPER, REVISION 17++

Free download at www.sanmayce.com; for Intel Merom 2.16GHz: 118KB/milisecc(EN) which is 262KB/milisecc(DE).

D:_KAZE_Kazuya_r17+_Lasse-Ariya-Oberhumer-Okumura (on_Osho)_32_bit_64_bit>Kazuya.exe/?
Kazuya(LZ Sentence_Dumper), revision 17+, written by Svalqyatchx,
in fact adapted from Lasse Reinhold's excellent QuickLZ 1.4.0 library,
in fact adapted from Ariya Hidayat's sub-excellent FastLZ 0.1.0 library,
in fact adapted from Markus F.X.J. Oberhumer's sub-excellent LZ0 2.03 library,
in fact adapted from Haruhiko Okumura's sub-excellent LZSS 4/6/1989 library.

Usage1: Kazuya [-sa1|-sa2|-sa3|-sa4|-SA1|-SA2|-SA3|-SA4
[-sa1|-sa2|-sa3|-sa4|-Sa1|-Sa2|-Sa3|-Sa4]-krknd] filename
Decodes all files from a list(filename)

-sa1 : QuickLZ Decode + Brute_Force Search Algorithm
-sa2 : QuickLZ Decode + Quick_Boyer_Moore Search Algorithm
-sa3 : QuickLZ Decode + SMITH_Boyer_Moore Search Algorithm
-sa4 : QuickLZ Decode + Karp_Rabin_Kaze Search Algorithm
-SA1 : LZ0 Decode + Brute_Force Search Algorithm
-SA2 : LZ0 Decode + Quick_Boyer_Moore Search Algorithm
-SA3 : LZ0 Decode + SMITH_Boyer_Moore Search Algorithm
-SA4 : LZ0 Decode + Karp_Rabin_Kaze Search Algorithm
-sa1 : FastLZ Decode + Brute_Force Search Algorithm
-sa2 : FastLZ Decode + Quick_Boyer_Moore Search Algorithm
-sa3 : FastLZ Decode + SMITH_Boyer_Moore Search Algorithm
-sa4 : FastLZ Decode + Karp_Rabin_Kaze Search Algorithm
-Sa1 : OkumuraLZ Decode + Brute_Force Search Algorithm
-sa2 : OkumuraLZ Decode + Quick_Boyer_Moore Search Algorithm
-sa3 : OkumuraLZ Decode + SMITH_Boyer_Moore Search Algorithm
-sa4 : OkumuraLZ Decode + Karp_Rabin_Kaze Search Algorithm
-krknd : Karp_Rabin_Kaze Search Algorithm, but with no additional
chunk-searches overhead and without decompression of
incoming files i.e. pure text is uploaded

Default is QuickLZ Decode + HORSPOOL_Boyer_Moore Search Algorithm

Usage2: Kazuya <e|d|E|D|A|R|a|r> inputFile outputFile

e: encode QuickLZ file
d: decode QuickLZ file
E: encode LZ0 file
D: decode LZ0 file
A: encode(archive) FastLZ file
R: decode(restore) FastLZ file
a: encode(archive) OkumuraLZ file
r: decode(restore) OkumuraLZ file

Example1: Kazuya Kazuya.lst

Example2: Kazuya -SA2 Kazuya.lst

Example3: Kazuya e Caterpillar.001.txt Caterpillar.001.txt.Lasse

Note1: Benchmark(HDD read speed is the nasty bottleneck):

Kazuya(EN:37KB/clock, DE:88KB/clock) for 24.9GB(10.6GB Lasse) texts.

Me machine is:

Motherboard Name: Toshiba Satellite L305
CPU Type: Mobile DualCore Intel Pentium, 2166 MHz (13 x 167)
CPU Alias: Merom-1M
L1 Code Cache: 32 KB per core
L1 Data Cache: 32 KB per core
L2 Cache: 1 MB (On-Die, ECC, ASC, Full-Speed)
Bus Type: Dual DDR2 SDRAM
Bus Width: 128-bit
Real Clock: 333 MHz (DDR)
Effective Clock: 666 MHz

Note2: Disastrous performance in case 256MB not fully physical!

Note3: Matches(hits) are overwritten to Kazuya.hits.Pattern?.html files.

Note4: works both on UNIX(LF) and windows(CRLF) text files.

Note5: Never forget the importance of defragmented_AND_grouped files located at
fastest area of disk - first partition is faster than second one, etc.

Note6: In ANSI, clock is defined as '#define CLOCKS_PER_SEC 1000'.

Note7: Since kazuya 13++:

- limits(just skip longer ones) lines to 960 chars; OTHERWISE: HUGE TIME
DELAYS due to recursive function;
- shows hits to console too; MORE VIVID;

Note8: Since Kazuya 14:

- No deletion of input file after compressing/decompressing;

Note9: During execution hitting a 'Esc' causes termination(i.e. skipping rest).

NoteA: This revision works with up to 127MB incoming(non-compressed) files,
because it allocates 256MB of which one half is for incoming
other for outgoing file, i.e. decompression/compression is RAM to RAM.

NoteB: Charge(delivery) performance combines upload and decode performance.

NoteC: The two examples below show the need of one additional wildcard in
order to match CR for windows texts; end of line is LF(as in UNIX):

Pattern(s) example: Pattern1: #####%
Pattern1_NestedPattern1:
Possible hit: NEW YORK
Pattern(s) example: Pattern1: \$\$\$\$\$\$\$@
Pattern1_NestedPattern1:
Possible hit: Printing

Pattern(s) note: You may specify(four times) a main-pattern(case insensitive
with wildcards '*' i.e. any character(s) or empty, also '@'
i.e. any character or empty, also '#' i.e. any character
and not empty, also '\$' i.e. any ALPHA character
and not empty, also '%' i.e. any NON-ALPHA character
and not empty) with three nested-patterns(case
sensitive and unexact), all four connected with AND.
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(s) example: Pattern1: *%take@\$\$\$@
Pattern1_NestedPattern1:
Possible hit: ... is taken by
Possible hit: ... would take it
Note: First % is to avoid e.g. 'mis' prefix
Second % is to avoid e.g. 'ing' suffix

Master-pattern note: It is case insensitive with wildcards '*', '@', '#', '\$', '%'
allowed. The purpose of this pattern is to
decide whether a search for next patterns will be
executed, it is applied on all lines i.e. the whole file.
There must be at least one hit in order to execute search
for next patterns.

Have a nice Kazuyaing.

For contacts: sanmayce@sanmayce.com

Sanmayce Svalqyatchx 'Kaze', 2010 May 24.

Allocated memory for DEncoded file in MB: 256

Kazuya: Can't open /? file.

D:_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>type Test_autoinput.txt

*said*as*as*
whenever

*to??
where

did
at?

*a*and*to*but*not*against*it*
again

D:_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>type Test_filelist_lasse.lst

Folio VIP - Osho's Books on CD-ROM Part 1.txt.lasse
Folio VIP - Osho's Books on CD-ROM Part 2.txt.lasse
Folio VIP - Osho's Books on CD-ROM Part 3.txt.lasse
Folio VIP - Osho's Books on CD-ROM Part 4.txt.lasse
Folio VIP - Osho's Books on CD-ROM Part 5.txt.lasse

D:_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>type Test_filelist_LZ0.lst

Folio VIP - Osho's Books on CD-ROM Part 1.txt.LZ0
Folio VIP - Osho's Books on CD-ROM Part 2.txt.LZ0
Folio VIP - Osho's Books on CD-ROM Part 3.txt.LZ0
Folio VIP - Osho's Books on CD-ROM Part 4.txt.LZ0
Folio VIP - Osho's Books on CD-ROM Part 5.txt.LZ0

D:_KAZE_Kazuya_r17_Test_Lasse-Oberhumer-Ariya_On_Osho_32_64_bit_NO-LZ0>type Test_filelist_FastLZ.lst

Folio VIP - Osho's Books on CD-ROM Part 1.txt.FastLZ
Folio VIP - Osho's Books on CD-ROM Part 2.txt.FastLZ
Folio VIP - Osho's Books on CD-ROM Part 3.txt.FastLZ
Folio VIP - Osho's Books on CD-ROM Part 4.txt.FastLZ

Folio VIP - Osho's Books on CD-ROM Part 5.txt.FastLZ

D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit>type
Test_filelist_OkumuraLZ.lst

Folio VIP - Osho's Books on CD-ROM Part 1.txt.Okumura
Folio VIP - Osho's Books on CD-ROM Part 2.txt.Okumura
Folio VIP - Osho's Books on CD-ROM Part 3.txt.Okumura
Folio VIP - Osho's Books on CD-ROM Part 4.txt.Okumura
Folio VIP - Osho's Books on CD-ROM Part 5.txt.Okumura

D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit>type Test_filelist_NON-
ENCODED.lst

Folio VIP - Osho's Books on CD-ROM Part 1.txt
Folio VIP - Osho's Books on CD-ROM Part 2.txt
Folio VIP - Osho's Books on CD-ROM Part 3.txt
Folio VIP - Osho's Books on CD-ROM Part 4.txt
Folio VIP - Osho's Books on CD-ROM Part 5.txt

D:_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>type Test_Lasse.bat

@echo off
Kazuya.exe -sa1 Test_filelist_lasse.lst<Test_autoinput.txt
echo.
type Test_Toshiba_Satellite_L305.txt
echo.
echo Run it some times with minimum applications started, press a key ...
pause>nul

D:_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>type Test_LZO.bat

@echo off
Kazuya.exe -SA1 Test_filelist_LZO.lst<Test_autoinput.txt
echo.
type Test_Toshiba_Satellite_L305.txt
echo.
echo Run it some times with minimum applications started, press a key ...
pause>nul

D:_KAZE_Kazuya_r17_Test_Lasse-Oberhumer-Ariya_On_Osho_32_64_bit_NO-LZO>type Test_FastLZ.bat

@echo off
Kazuya.exe -sa1 Test_filelist_FastLZ.lst<Test_autoinput.txt
echo.
type Test_Toshiba_Satellite_L305.txt
echo.
echo Run it some times with minimum applications started, press a key ...
pause>nul

D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit>type Test_OkumuraLZ.bat

@echo off
Kazuya.exe -sa1 Test_filelist_OkumuraLZ.lst<Test_autoinput.txt
echo.
type Test_Toshiba_Satellite_L305.txt
echo.
echo Run it some times with minimum applications started, press a key ...
pause>nul

D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit>type Test_NON-ENCODED.bat

@echo off
Kazuya.exe -krknd Test_filelist_NON-ENCODED.lst<Test_autoinput.txt
echo.
type Test_Toshiba_Satellite_L305.txt
echo.
echo Run it some times with minimum applications started, press a key ...
pause>nul

D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit>dir/og/on/s

Directory of D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit

02/15/2010	04:01 AM	<DIR>	Backgrounds
02/15/2010	05:35 AM	<DIR>	Kazuya17
02/14/2010	03:56 AM		42,676,189 Folio VIP - Osho's Books on CD-ROM Part 1.txt
02/14/2010	03:45 AM		20,158,572 Folio VIP - Osho's Books on CD-ROM Part 1.txt.FastLZ
02/14/2010	06:47 AM		16,330,610 Folio VIP - Osho's Books on CD-ROM Part 1.txt.lasse

02/14/2010	04:24 AM		20,037,516 Folio VIP - Osho's Books on CD-ROM Part 1.txt.LZO
02/14/2010	11:17 PM		18,233,128 Folio VIP - Osho's Books on CD-ROM Part 1.txt.Okumura
02/14/2010	03:56 AM		41,123,931 Folio VIP - Osho's Books on CD-ROM Part 2.txt
02/14/2010	03:46 AM		19,518,872 Folio VIP - Osho's Books on CD-ROM Part 2.txt.FastLZ
02/14/2010	06:47 AM		15,736,568 Folio VIP - Osho's Books on CD-ROM Part 2.txt.lasse
02/14/2010	04:25 AM		19,303,785 Folio VIP - Osho's Books on CD-ROM Part 2.txt.LZO
02/14/2010	11:17 PM		17,761,259 Folio VIP - Osho's Books on CD-ROM Part 2.txt.Okumura
02/14/2010	03:57 AM		42,447,586 Folio VIP - Osho's Books on CD-ROM Part 3.txt
02/14/2010	03:46 AM		19,878,910 Folio VIP - Osho's Books on CD-ROM Part 3.txt.FastLZ
02/14/2010	06:47 AM		16,064,789 Folio VIP - Osho's Books on CD-ROM Part 3.txt.lasse
02/14/2010	04:25 AM		19,786,666 Folio VIP - Osho's Books on CD-ROM Part 3.txt.LZO
02/14/2010	11:17 PM		17,983,244 Folio VIP - Osho's Books on CD-ROM Part 3.txt.Okumura
02/14/2010	03:57 AM		41,875,240 Folio VIP - Osho's Books on CD-ROM Part 4.txt
02/14/2010	03:46 AM		19,511,239 Folio VIP - Osho's Books on CD-ROM Part 4.txt.FastLZ
02/14/2010	06:47 AM		15,735,147 Folio VIP - Osho's Books on CD-ROM Part 4.txt.lasse
02/14/2010	04:25 AM		19,342,946 Folio VIP - Osho's Books on CD-ROM Part 4.txt.LZO
02/14/2010	11:18 PM		17,659,991 Folio VIP - Osho's Books on CD-ROM Part 4.txt.Okumura
02/14/2010	03:57 AM		38,786,003 Folio VIP - Osho's Books on CD-ROM Part 5.txt
02/14/2010	03:46 AM		18,169,086 Folio VIP - Osho's Books on CD-ROM Part 5.txt.FastLZ
02/14/2010	06:47 AM		14,765,031 Folio VIP - Osho's Books on CD-ROM Part 5.txt.lasse
02/14/2010	04:25 AM		18,043,396 Folio VIP - Osho's Books on CD-ROM Part 5.txt.LZO
02/14/2010	11:18 PM		16,304,888 Folio VIP - Osho's Books on CD-ROM Part 5.txt.Okumura
02/15/2010	04:44 AM		90,112 Kazuya.exe
02/15/2010	03:56 AM		2,128,897 Kazuya.gif
02/15/2010	04:56 AM		1,300 Kazuya.hits.pattern1.html
02/15/2010	04:56 AM		10,293 Kazuya.hits.pattern2.html
02/15/2010	04:56 AM		9,089 Kazuya.hits.pattern3.html
02/15/2010	04:56 AM		538 Kazuya.hits.pattern4.html
02/15/2010	04:43 AM		135,680 Kazuya_x64.exe
02/14/2010	06:47 AM		491,414 Report_EVEREST Ultimate Edition_Test_Toshiba_Satellite_L305.htm
02/15/2010	05:01 AM		525,301 Report_EVEREST Ultimate Edition_Test_Toshiba_Satellite_L305.png
02/14/2010	06:47 AM		96 Test_autoinput.txt
02/14/2010	06:47 AM		210 Test_FastLZ.bat
02/14/2010	06:47 AM		270 Test_filelist_FastLZ.lst
02/14/2010	06:47 AM		265 Test_filelist_lasse.lst
02/14/2010	06:47 AM		255 Test_filelist_LZO.lst
02/15/2010	01:38 AM		235 Test_filelist_NON-ENCODED.lst
02/15/2010	12:22 AM		275 Test_filelist_OkumuraLZ.lst
02/14/2010	06:47 AM		209 Test_Lasse.bat
02/14/2010	06:47 AM		207 Test_LZO.bat
02/15/2010	01:41 AM		217 Test_NON-ENCODED.bat
02/15/2010	04:36 AM		667,768 Test_NON-ENCODED_Kazuya.hits.pattern3.html.png
02/15/2010	12:21 AM		213 Test_OkumuraLZ.bat
02/15/2010	04:04 AM		1,355 Test_Toshiba_Satellite_L305.txt
		48 File(s)	571,299,035 bytes

Directory of D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit\Backgrounds

02/15/2010	03:52 AM		2,150,463 Caterpillar.gif
02/15/2010	03:43 AM		2,339,680 GRID_TILE-READY.png
02/15/2010	03:51 AM		27,763,386 GRID_TILE-READY_Caterpillar.psd
02/15/2010	03:46 AM		27,677,539 GRID_TILE-READY_Kazuya.psd
02/15/2010	03:54 AM		27,846,963 GRID_TILE-READY_salah-ed-din.psd
02/15/2010	03:55 AM		2,167,397 salah-ed-din.gif
		6 File(s)	89,945,428 bytes

Directory of D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit\Kazuya17

02/14/2010	05:56 AM	<DIR>	_Others_LZs
02/14/2010	06:47 AM		259 _CompileWINx32_faster2003compiler.bat
02/14/2010	06:47 AM		733 _CompileWINx64.bat
02/14/2010	06:47 AM		14,201 fastlz.c
02/14/2010	06:47 AM		3,666 fastlz.h
02/14/2010	06:47 AM		4,286 k32_.ico
02/14/2010	06:47 AM		4,286 k64_.ico
02/15/2010	04:44 AM		225,037 Kazuya.asm
02/15/2010	04:43 AM		116,106 Kazuya.c
02/15/2010	04:44 AM		90,112 Kazuya.exe
02/15/2010	04:43 AM		135,680 Kazuya_x64.exe
02/15/2010	04:43 AM		212,924 Kazuya_x64_blend.asm
02/14/2010	06:47 AM		212,184 KAZUYA18_DOS.ZIP
02/14/2010	06:47 AM		1,010,688 Kazuyalogo.doc

```

02/14/2010 06:47 AM      594,851 Kazuyalogo_diz_.pdf
02/15/2010 05:35 AM      76,800 Kazuyalogo_test.doc
02/14/2010 06:47 AM      14,090 lzoconf.h
02/14/2010 06:47 AM      65,826 lzdefs.h
02/15/2010 04:44 AM      35,754 minilzo.asm
02/14/2010 06:47 AM      124,929 minilzo.c
02/14/2010 06:47 AM        3,493 minilzo.h
02/15/2010 04:43 AM      36,332 minilzo_x64_blend.asm
02/15/2010 04:44 AM      21,652 quicklz.asm
02/14/2010 06:47 AM      20,292 quicklz.c
02/14/2010 06:47 AM        2,971 quicklz.h
02/15/2010 04:43 AM      21,658 quicklz_x64_blend.asm
      25 File(s)      3,048,810 bytes

```

Directory of D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit\Kazuya17_Others_LZs

```

02/14/2010 05:56 AM <DIR>      minilzo-2.03
02/14/2010 06:47 AM      59,741 liblzf-3.5.tar.gz
02/14/2010 06:47 AM      626,042 lzo-2.03.tar.gz
      2 File(s)      685,783 bytes

```

Directory of D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit\Kazuya17_Others_LZs\minilzo-2.03

```

02/14/2010 06:47 AM      14,090 lzoconf.h
02/14/2010 06:47 AM      65,826 lzdefs.h
02/14/2010 06:47 AM      124,929 minilzo.c
02/14/2010 06:47 AM        3,493 minilzo.h
02/14/2010 06:47 AM        4,053 README.LZO
02/14/2010 06:47 AM        5,549 testmini.c
      6 File(s)      217,940 bytes

```

Total Files Listed:
87 File(s) 665,196,996 bytes

D:_KAZE_Kazuya_r17_Lasse-Ariya-Oberhumer-Okumura_(on_Osho)_32_bit_64_bit>

QuickLZ(for 206,908,949 bytes real-world text compressed to **78,632,145** bytes):

```

Total Upload performance: 000,808KB/clock
Total Decode performance: 000,122KB/clock(EN) or 000,323KB/clock(DE)
Total Charge performance: 000,106KB/clock(EN) or 000,280KB/clock(DE)
Total Upload time: 95 clocks
Total Decode time: 625 clocks

```

FastLZ(for 206,908,949 bytes real-world text compressed to **97,236,679** bytes):

```

Total Upload performance: 000,863KB/clock
Total Decode performance: 000,121KB/clock(EN) or 000,258KB/clock(DE)
Total Charge performance: 000,106KB/clock(EN) or 000,226KB/clock(DE)
Total Upload time: 110 clocks
Total Decode time: 781 clocks

```

LZO(for 206,908,949 bytes real-world text compressed to **96,514,309** bytes):

```

Total Upload performance: 001,013KB/clock
Total Decode performance: 000,111KB/clock(EN) or 000,239KB/clock(DE)
Total Charge performance: 000,100KB/clock(EN) or 000,215KB/clock(DE)
Total Upload time: 93 clocks
Total Decode time: 844 clocks

```

OkumuraLZ(for 206,908,949 bytes real-world text compressed to **87,942,510** bytes):

```

Total Upload performance: 000,913KB/clock
Total Decode performance: 000,077KB/clock(EN) or 000,182KB/clock(DE)
Total Charge performance: 000,071KB/clock(EN) or 000,167KB/clock(DE)
Total Upload time: 94 clocks
Total Decode time: 1,110 clocks

```

```

D:\_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>type Kazuya.hits.pattern1.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><html><head><title>Kazuya
Hits</title><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head><body
style="background-image: url(./Kazuya.gif); color: rgb(0, 200, 221); background-color: rgb(35, 35,
35);"alink="#003366" link="#006699" vlink="#003366"><font color="#7788AA" face="Courier New, Courier,
mono"><pre>

```

```

    In fact, Buddha has said that whenever a Buddha is born, not only do the past Buddhas feel it,
    He said, "I am as much surprised as you are. I don't know. And whenever I try to do it, it never
    whenever he said, "You go to the market and you start chakallas," the whole class would laugh.
    was time to rest. But whenever the young man said it was time to rest, he was always willing to rest
    It is said that whenever Gautam Buddha passed by any path, even if it was not the time for the
    The monk said, "It is difficult, because Buddha has said that whenever you ask for something,
    much, whenever he passed through a village he said that tathata is passing. It means one who has
    It is said that whenever Mark Twain was asked a question he always replied with another

```

```
</pre></font></body></html>
```

```

D:\_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>type Kazuya.hits.pattern4.html

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><html><head><title>Kazuya
Hits</title><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head><body
style="background-image: url(./Kazuya.gif); color: rgb(0, 200, 221); background-color: rgb(35, 35,
35);"alink="#003366" link="#006699" vlink="#003366"><font color="#7788AA" face="Courier New, Courier,
mono"><pre>

```

```

    transcended, and it is a great experience to transcend sex, but it cannot be done if you are against it.

```

```
</pre></font></body></html>
```

```

D:\_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>type Test Toshiba Satellite L305.txt

```

```

# On Toshiba Satellite L305 laptop computer for following functions:
# - Decompression: (QuickLZ or LZO or FastLZ or OkumuraLZ) RAM to RAM;
# - Search: Brute-force, Wildcard, Skip, Hash;
# with Mobile Intel Pentium T3400 Merom-1M 2166MHz CPU results are:
#
# Total Upload performance: 000,711KB/clock
# Total Decode performance: 000,119KB/clock(EN) or 000,314KB/clock(DE)
# Total Charge performance: 000,102KB/clock(EN) or 000,269KB/clock(DE)
# Total Upload time: 108 clocks
# Total Decode time: 642 clocks
# Total Search time: 6,640 clocks
# Total time: 8 second(s)
# Total bytes decoded: 206,908,949
# Total Search(non-mask) function invocations: 9,838,032
# Total MASK i.e. wildcard invocations: 60,822
# Total MASK i.e. wildcard hits: 195
# Total MASK i.e. wildcard time: 498 clocks
# Total MASK i.e. wildcard performance: 12KB/clock
# Total BoyerMooreHorspool invocations: 20
# Total BoyerMooreHorspool(whole chunks, not lines) hits: 70,913
# Total BoyerMooreHorspool(whole chunks, not lines) time: 827 clocks
# Total BoyerMooreHorspool(whole chunks, not lines) performance: 977KB/clock
# Total KarpRabinKaze invocations: 20
# Total KarpRabinKaze(whole chunks, not lines) hits: 70,913
# Total KarpRabinKaze(whole chunks, not lines) time: 1,238 clocks
# Total KarpRabinKaze(whole chunks, not lines) performance: 652KB/clock

```

D:_KAZE_Kazuya_r16_Test_Lasse-Oberhumer_On_Osho>

QuickLZ EVEREST v5.00.1650 Memory Copy: Pentium T3400 2166 MHz; Toshiba Satellite L305: GL40 Int.; Dual DDR2-667: 5-5-5-13: 3725MB/s;

Total Decode performance: 000,122KB/clock(EN) or 000,323KB/clock(DE)

FastLZ EVEREST v5.00.1650 Memory Copy: Pentium T3400 2166 MHz; Toshiba Satellite L305: GL40 Int.; Dual DDR2-667: 5-5-5-13: 3725MB/s;

Total Decode performance: 000,121KB/clock(EN) or 000,258KB/clock(DE)

LZO EVEREST v5.00.1650 Memory Copy: Pentium T3400 2166 MHz; Toshiba Satellite L305: GL40 Int.; Dual DDR2-667: 5-5-5-13: 3725MB/s;

Total Decode performance: 000,111KB/clock(EN) or 000,239KB/clock(DE)

OkumuraLZ EVEREST v5.00.1650 Memory Copy: Pentium T3400 2166 MHz; Toshiba Satellite L305: GL40 Int.; Dual DDR2-667: 5-5-5-13: 3725MB/s;

Total Decode performance: 000,077KB/clock(EN) or 000,182KB/clock(DE)

QuickLZ _____ :

Total Decode performance: 000, _____ KB/clock(EN) or 000, _____ KB/clock(DE)

FastLZ _____ :

Total Decode performance: 000, _____ KB/clock(EN) or 000, _____ KB/clock(DE)

LZO _____ :

Total Decode performance: 000, _____ KB/clock(EN) or 000, _____ KB/clock(DE)

OkumuraLZ _____ :

Total Decode performance: 000, _____ KB/clock(EN) or 000, _____ KB/clock(DE)

QuickLZ _____ :

Total Decode performance: 000, _____ KB/clock(EN) or 000, _____ KB/clock(DE)

FastLZ _____ :

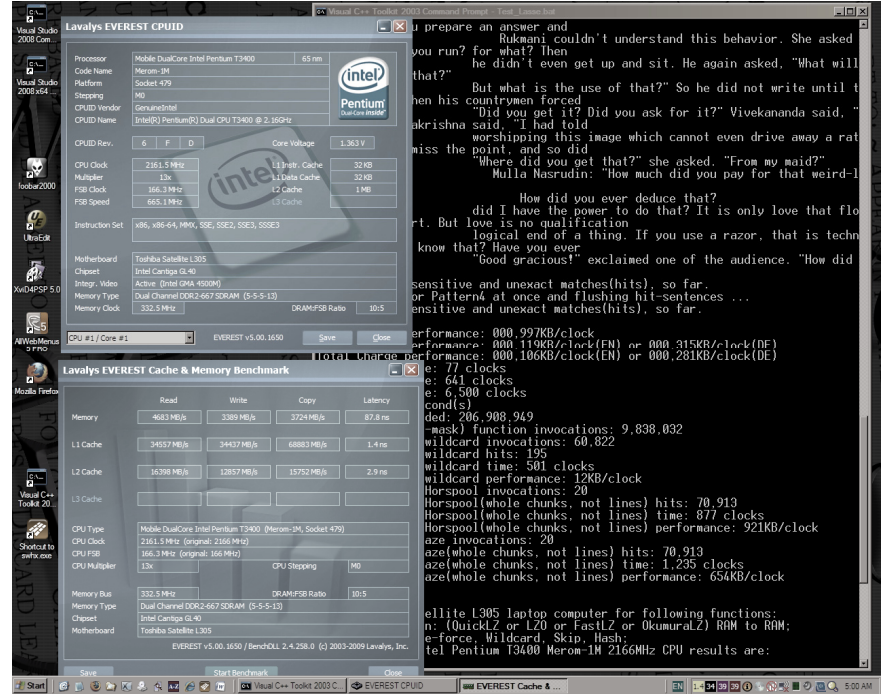
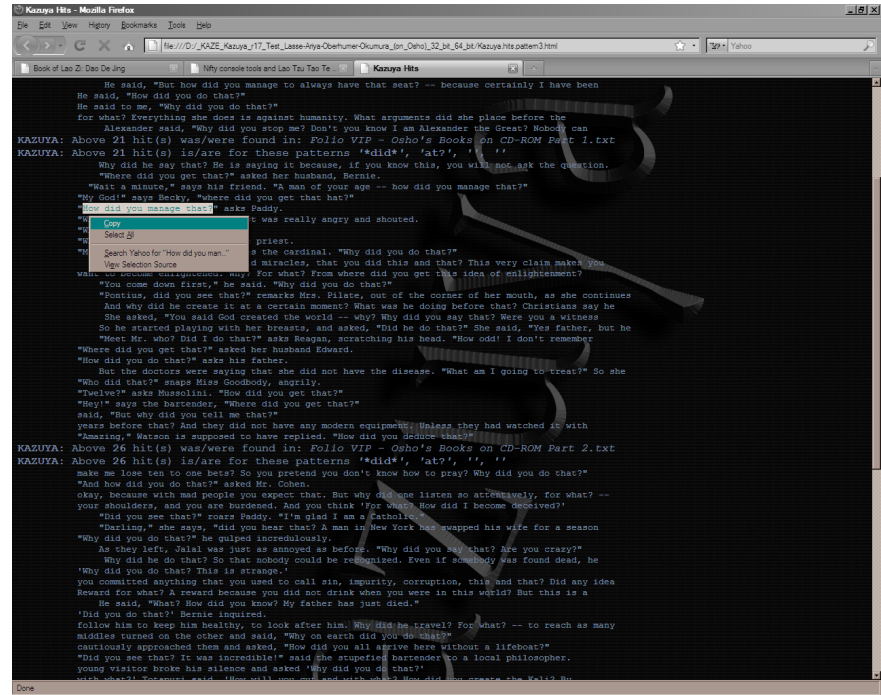
Total Decode performance: 000, _____ KB/clock(EN) or 000, _____ KB/clock(DE)

LZO _____ :

Total Decode performance: 000, _____ KB/clock(EN) or 000, _____ KB/clock(DE)

OkumuraLZ _____ :

Total Decode performance: 000, _____ KB/clock(EN) or 000, _____ KB/clock(DE)



Visual C++ Toolkit 2003 Command Prompt - kazuya-krknd q.lst

```

C:\WorkTemp\balz-1.15_QUAD-1.12\Osho's Books_test>type q.lst
dec.txt
Folio VIP - Osho's Books on CD-ROM Part 1-2-3-4-5.txt
HINIS.TXT
LHA255.TXT
license.txt
lzturbo.txt
OSHO.TXT
README.TXT
REVISION.TXT

C:\WorkTemp\balz-1.15_QUAD-1.12\Osho's Books_test>kazuya-krknd q.lst
Kazuya(LZ Sentence_Dumper), revision 17+ written by Svalgwatck,
in fact adapted from Lasse Reinhold's excellent QuickLZ 1.4.0 library,
in fact adapted from Ariya Hidayat's sub-excellent FastLZ 0.1.0 library,
in fact adapted from Markus F.X.J. Oberhumer's sub-excellent LZ0 2.03 library,
in fact adapted from Haruhiko Okumura's sub-excellent LZSS 4/6/1989 library.

Usage1: Kazuya [-sa1] [-sa2] [-sa3] [-sa4] [-S01] [-S02] [-S03] [-S04]
              [-sH1] [-sH2] [-sH3] [-sH4] [-Sa1] [-Sa2] [-Sa3] [-Sa4] [-krknd] filename
Decodes all files from a list(filename)
-sa1: QuickLZ Decode + Brute Force Search Algorithm
-sa2: QuickLZ Decode + Quick Boyer Moore Search Algorithm
-sa3: QuickLZ Decode + SMITH Boyer Moore Search Algorithm
-sa4: QuickLZ Decode + Karp_Rabin_Kaze Search Algorithm
-S01: LZ0 Decode + Brute Force Search Algorithm
-S02: LZ0 Decode + Quick Boyer Moore Search Algorithm
-S03: LZ0 Decode + SMITH Boyer Moore Search Algorithm
-S04: LZ0 Decode + Karp_Rabin_Kaze Search Algorithm
-sH1: FastLZ Decode + Brute Force Search Algorithm
-sH2: FastLZ Decode + Quick Boyer Moore Search Algorithm
-sH3: FastLZ Decode + SMITH Boyer Moore Search Algorithm
-sH4: FastLZ Decode + Karp_Rabin_Kaze Search Algorithm
-krknd: Karp_Rabin_Kaze Search Algorithm, but with no additional
chunk-searches overhead and without decompression of
incoming files i.e. pure text is uploaded
Default is QuickLZ Decode + HORSPOOL Boyer Moore Search Algorithm
Usage2: Kazuya [e|d|H|R|a|r] inputFile outputFile
e: encode QuickLZ file
d: decode QuickLZ file
E: encode LZ0 file
D: decode LZ0 file
R: encode(archive) FastLZ file
r: decode(restore) FastLZ file
a: encode(archive) OkumuraLZ file
r: decode(restore) OkumuraLZ file
Example1: Kazuya Kazuya.lst
Example2: Kazuya -S02 Kazuya.lst
Example3: Kazuya -e Caterpillar.001.txt Caterpillar.001.txt.Lasse

```

Visual C++ Toolkit 2003 Command Prompt

```

NoteC: The two examples below show the need of one additional wildcard in
order to match CR for windows texts; end of line is LF(as in UNIX):
Pattern(s) example: Pattern: #####
Possible hit: #####
Pattern(s) example: Pattern: #####$
Possible hit: #####
Pattern(s) note: You may specify(four times) a main-pattern(case insensitive
with wildcards "*" i.e. any character(s) or empty, also "@"
i.e. any character or empty, also "!" i.e. any character
and not empty, also "$" i.e. any ALPHA character
and not empty, also "%" i.e. any NON-ALPHA character
and not empty) with three nested-patterns(case
sensitive and unexact) all four connected with AND
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(s) example: Pattern: *%take@$$%
Input Pattern_NestedPattern:
Possible hit: ... is taken by
Possible hit: ... would take it
Note: First % is to avoid e.g. 'mis' prefix
Second % is to avoid e.g. 'ing' suffix
Master-pattern note: It is case insensitive with wildcards "*", "@", "!", "$", "%"
allowed, the purpose of this pattern is to
decide whether a search for next patterns will be
executed, it is applied on all lines i.e. the whole file.
There must be at least one hit in order to execute search
for next patterns.

Have a nice Kazuyaing.
For contacts: sanmayce@sanmayce.com
Sanmayce Svalgwatck 'Kaze', 2010 May 24.

Allocated memory for DEcoded file in MB: 256
Size of input file with files for Kazuyaing: 149

Input Master-pattern(hit only 'Enter' to skip): *%lzturbo%
Input Pattern(hit only 'Enter' to skip): *
Input Pattern_NestedPattern(hit only 'Enter' to skip):
Input Pattern2(hit only 'Enter' to skip):
Processing dec.txt ...
Overall upload performance so far: 000.000KB/clock
Doing SEARCH for Master-pattern ...
Processing Folio VIP - Osho's Books on CD-ROM Part 1-2-3-4-5.txt ...
Overall upload performance so far: 000.678KB/clock
Doing SEARCH for Master-pattern ...
Processing HINIS.TXT ...
Overall upload performance so far: 000.678KB/clock
Doing SEARCH for Master-pattern ...
Processing LHA255.TXT

```

Visual C++ Toolkit 2003 Command Prompt

```

000.000.051 lzturbo -59 sfc\* backup_dir
000.000.052 lzturbo -59 -r mydir\* backup_dir
000.000.053 lzturbo -59 -r c:\users\myusr\mydir
000.000.054
000.000.055 Decompression examples:
000.000.056 lzturbo -d enwik9.lzt restore_dir
000.000.057 lzturbo -d *.lzt restore_dir
000.000.058 lzturbo -d -r backup_dir\* restor
000.000.059
000.000.060 replace '\ ' for linux.
000.000.061
000.000.062
000.000.063
000.000.064 lzturbo uses ~15N/2N Mb memory in level 9
000.000.065 ion per processor/core (N=block size, default=32mb).
000.000.066 recommended options for benchmarking:
000.000.067 -10,-11 : compression time bench
000.000.068 -21,-31 : compression time
000.000.069 -19,-29,-39 : decompression time be
000.000.070 -43 : Efficiency benchmarks
000.000.071
000.000.072 FILES:
000.000.073 lzturbo.exe : win32.
000.000.074 lzturbo_ocpu.exe : win32 (use only whe
000.000.075
000.000.076 lzturbo : 32-bits lin
000.000.077 lzturbo_static : 32-bits lin
000.000.078
000.000.079 lzturbo.txt: this file
and 19 cases insensitive and unexact matches(hits).
Processing OSHO.TXT ...
Overall upload performance so far: 000.737KB/clock
Doing SEARCH for Master-pattern ...
Processing README.TXT ...
Overall upload performance so far: 000.737KB/clock
Doing SEARCH for Master-pattern ...
Processing REVISION.TXT ...
Overall upload performance so far: 000.737KB/clock
Doing SEARCH for Master-pattern ...

Total Upload performance: 000.737KB/clock
Total Upload time: 548 clocks
Total Search time: 10,546 clocks
Total time: 11 seconds!
Total bytes decoded: 413,891,925
Total Search(non-mask) function invocations: 0
Kazuya: Done.

```

Visual C++ Toolkit 2003 Command Prompt

```

Note: First % is to avoid e.g. 'mis' prefix
Second % is to avoid e.g. 'ing' suffix
Master-pattern note: It is case insensitive with wildcards "*", "@", "!", "$", "%"
allowed, the purpose of this pattern is to
decide whether a search for next patterns will be
executed, it is applied on all lines i.e. the whole file.
There must be at least one hit in order to execute search
for next patterns.

Have a nice Kazuyaing.
For contacts: sanmayce@sanmayce.com
Sanmayce Svalgwatck 'Kaze', 2010 May 24.

Allocated memory for DEcoded file in MB: 256
Size of input file with files for Kazuyaing: 149

Input Master-pattern(hit only 'Enter' to skip): *%sho%
Input Pattern(hit only 'Enter' to skip): *%take@$$%
Input Pattern_NestedPattern(hit only 'Enter' to skip):
Input Pattern2(hit only 'Enter' to skip):

```