

MASAKARI-The_Holy_Axe_Textual_Sidekick



Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```
00,001 ' Masakari.bas, written in QB64 by Kaze (sanmayce@sanmayce.com), 2021-Mar-03
00,002 ' Revised 2021-Jul-28, fixed double-clicks, fixes in combo handling; fixed report line length bug when (ExpandTabs was not dealing with the full size).
00,003 ' Many thanks go to Galleon, also thanks go to the www.qb64.org/forum members for sharing useful excerpts/etudes
00,004
00,005 ' Note #1: For wrapping set the flag 'WrapFlag = 1' or 'WrapFlag = 0' for vanilla.
00,006 ' 'MASAKARI_General-Purpose_Grade_English_Wordlist.wrd' is the unigram i.e. wordlist to be used here.
00,007
00,008 CONST RSHIFTkey& = 100303
00,009 CONST LSHIFTkey& = 100304
00,010 CONST RCTRLkey& = 100305
00,011 CONST LCTRLkey& = 100306
00,012 CONST RALTkey& = 100307
00,013 CONST LALTkey& = 100308
00,014
00,015 CONST BACKSPCkey& = 8
00,016 CONST TABkey& = 9
00,017 CONST SPACEkey& = 32
00,018 CONST ESCkey& = 27
00,019 CONST ENTERkey& = 13
00,020
00,021 CONST HOMEkey& = 18176
00,022 CONST ENDkey& = 20224
00,023
00,024 CONST INSkey& = 20992
00,025 CONST DELkey& = 21248
00,026
00,027 CONST PGUPkey& = 18688
00,028 CONST PGDNkey& = 20736
00,029
00,030 CONST LEFTkey& = 19200
00,031 CONST RIGHTkey& = 19712
00,032 CONST UPkey& = 18432
00,033 CONST DOWNkey& = 20480
00,034
00,035 DIM SHARED ToLoadOrNotFlag AS INTEGER ' Global flag, if 1 then load the entire file - use it to speed up parsing. Set to 0 if you want to load bigger files, slow it is.
00,036 DIM SHARED WrapFlag AS INTEGER
00,037
00,038 _DEFINE A-Z AS _INTEGER64
00,039
00,040 'Set compile-time codepaths:
00,041 'Vanilla (WRAPPERV = 0) or Wrapper (WRAPPERV = 1):
00,042 $LET WRAPPERV = 0
00,043 'Toshiba 8x16 (BIGORSMALL = 1) or NEC 6x12 (BIGORSMALL = 0):
00,044 $LET BIGORSMALL = 1
00,045
00,046 $IF WRAPPERV = 0 THEN
00,047     WrapFlag = 0 ' 1 means wrapping; 0 means vanilla
00,048 $ELSEIF WRAPPERV = 1 THEN
00,049     WrapFlag = 1 ' 1 means wrapping; 0 means vanilla
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,050 $ELSE
00,051 $END IF
00,052
00,053 FONT4x = 0 '32px in 98x30
00,054 ToLoadOrNotFlag = 0 ' Should be 0; OLD: 1 means fast load but memory greedy; 0 means slow load but memory efficient
00,055
00,056 ShutDownDuration# = 4 ' 4 seconds
00,057 DclickTime# = 0.33 ' 1/3 of a second, usually it is 0.27s, so 0.33 suits even the slow clickers
00,058 DIM Button1LOG_firstDetection#(4)
00,059 DIM Button1LOG_ForHowLongHoled#(4)
00,060 DIM Button2LOG_firstDetection#(4)
00,061 DIM Button2LOG_ForHowLongHoled#(4)
00,062 PrevClick1# = 0
00,063 PrevClick2# = 0
00,064
00,065 DIM KeyTap_Button1LOG_firstDetection#(4)
00,066 DIM KeyTap_Button1LOG_ForHowLongHoled#(4)
00,067 DIM KeyTap_Button2LOG_firstDetection#(4)
00,068 DIM KeyTap_Button2LOG_ForHowLongHoled#(4)
00,069 DIM KeyTap_Button3LOG_firstDetection#(4)
00,070 DIM KeyTap_Button3LOG_ForHowLongHoled#(4)
00,071 DIM KeyTap_Button4LOG_firstDetection#(4)
00,072 DIM KeyTap_Button4LOG_ForHowLongHoled#(4)
00,073 KeyTap_PrevClick1# = 0
00,074 KeyTap_PrevClick2# = 0
00,075 KeyTap_PrevClick3# = 0
00,076 KeyTap_PrevClick4# = 0
00,077
00,078 _CONTROLCHR OFF 'without it couldn't _MAPUNICODE these 7, 9..13, 28..31
00,079
00,080 IF INSTR(LCASE$(COMMAND$), "/help") OR INSTR(LCASE$(COMMAND$), "-h") THEN
00,081     $SCREENHIDE
00,082     $CONSOLE
00,083     _CONSOLE ON
00,084     _CONSOLETITLE "Masakari console window"
00,085     _DEST _CONSOLE
00,086
00,087     ShowASCIIart
00,088
00,089     '
00,090     '  _
00,091     '  |/' |
00,092     '  /| |
00,093     '  \| /
00,094     '  _
00,095     '
00,096     '  /| |
00,097     '  /| |
00,098     '  | |

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

00,099 ' | |
00,100 '└─┘
00,101 ' | |
00,102 '└─┘
00,103 '└─┘
00,104 '└─┘
00,105 '└─┘
00,106 '└─┘
00,107 ' | |
00,108 '└─┘
00,109 '└─┘
00,110 '└─┘
00,111 '└─┘
00,112 '└─┘
00,113 ' | |
00,114 '└─┘
00,115 '└─┘
00,116 '└─┘
00,117 '└─┘
00,118 '└─┘
00,119 ' | |
00,120 '└─┘
00,121 '└─┘
00,122 '└─┘
00,123 '└─┘
00,124 '└─┘
00,125 ' | |
00,126 '└─┘
00,127 '└─┘
00,128 '└─┘
00,129 '└─┘
00,130 '└─┘
00,131 ' | |
00,132 '└─┘
00,133 '└─┘
00,134 '└─┘
00,135 '└─┘
00,136 '└─┘
00,137 ' | |
00,138 '└─┘
00,139 '└─┘
00,140 '└─┘
00,141 '└─┘
00,142 '└─┘
00,143 ' | |
00,144 '└─┘
00,145 '└─┘
00,146 '└─┘
00,147 '└─┘

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2 8x16.ttf; Downloadable at: www.sarmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```
00,197 END IF
00,198
00,199 ' _DEFINE A-Z AS _INTEGER64 'commented and moved up since the above defined 'FONT4x = 1' was seen as 0 below?!
00,200
00,201 $IF WINDOWS THEN
00,202     DECLARE LIBRARY "mem"
00,203         FUNCTION MemInUsePercent&&
00,204         FUNCTION TotalPhysicalMem&&
00,205         FUNCTION FreePhysicalMem&&
00,206         FUNCTION TotalPagingFile&&
00,207         FUNCTION FreePagingFile&&
00,208         FUNCTION TotalVirtualMem&&
00,209         FUNCTION FreeVirtualMem&&
00,210         FUNCTION FreeExtendedMem&&
00,211         FUNCTION GetCPULoad#
00,212     END DECLARE
00,213 $END IF
00,214
00,215 DIM SHARED Mwholefile AS _MEM
00,216 DIM SHARED MhandleOFF AS _MEM
00,217 DIM SHARED MhandleLEN AS _MEM 'paired with above one, added for speed, no need to search for the length at offset 'MhandleOFF'
00,218
00,219 DIM QWORD AS _UNSIGNED _INTEGER64
00,220 'DIM SHARED Byte AS STRING * 1 'not used since r.7
00,221 DIM PrevByte AS STRING * 1
00,222 DIM LastByte AS STRING * 1
00,223
00,224 _SCREENMOVE 102, 3
00,225 XdimCOL = 128
00,226 YdimROW = 40 ' ensure old laptops with 768pixels vertical will hold the whole window
00,227 wide& = _DESKTOPWIDTH
00,228 high& = _DESKTOPHEIGHT
00,229 IF high& > 1000 THEN YdimROW = 60
00,230 IF wide& >= 1920 THEN XdimCOL = 198
00,231
00,232 IF XdimCOL = 128 AND YdimROW = 40 THEN _TITLE "MASAKARI, revision 8.1+ 128x40, The 'Holy Axe' English Text Sidekick"
00,233 IF XdimCOL = 128 AND YdimROW = 60 THEN _TITLE "MASAKARI, revision 8.1+ 128x60, The 'Holy Axe' English Text Sidekick"
00,234 IF XdimCOL = 198 AND YdimROW = 60 THEN _TITLE "MASAKARI, revision 8.1+ 198x60, The 'Holy Axe' English Text Sidekick"
00,235
00,236 Mode4K = 0 'Make it 1 or 0, if 1 then 16 becomes 32 i.e. the font is 4x bigger: 8x16 becomes 16x32
00,237 IF wide& >= 1920 * 2 THEN Mode4K = 1
00,238
00,239 IF FONT4x = 1 THEN
00,240     XdimCOL = 112
00,241     YdimROW = 30 ' ensure old laptops with 768pixels vertical will hold the whole window
00,242     IF XdimCOL = 112 AND YdimROW = 30 THEN _TITLE "MASAKARI, revision 8.1+ 112x30, The 'Holy Axe' English Text Sidekick"
00,243 END IF
00,244
00,245 'Mode4K = 1: YdimROW = YdimROW \ 2: XdimCOL = 96 'debug; Enforcing 32px on 1680 or FHD is good, however status line should be shrunk, also Search Panel need 40lines not 30!
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

00,246
00,247 ASCIIFrame = 0
00,248 ASCIIFramePEN = 0
00,249 'YdimROW = 40 'debug, test the Search Panel
00,250
00,251 $IF WIN THEN
00,252     CONST Slash = "\"
00,253 $ELSE
00,254     CONST Slash = "/"
00,255 $END IF
00,256
00,257 $IF WINDOWS THEN
00,258     PSPlike$ = _CWD$ + "\"
00,259 $ELSE
00,260     PSPlike$ = _CWD$ + "/"
00,261 $END IF
00,262
00,263 'Note1: If the Toshiba (MxPlus_ToshibaTxL2_8x16.ttf) font is not in the starting folder then JPN 6x12 (Px437_DOS-V_re_JPN12.ttf) is loaded instead (if not 4K). On old laptops e.g. 1366x768 it can show the "standard
128x60".
00,264 'Note2: Trying to enforce the maximum, if (4K is 0) and (Toshiba is missing):
00,265 $IF BIGORSMALL = 1 THEN
00,266     IF _FILEEXISTS(PSPlike$ + "MxPlus_ToshibaTxL2_8x16.ttf") = 0 THEN
00,267         IF Mode4K = 0 THEN
00,268             IF wide& >= 1920 THEN
00,269                 XdimCOL = 300 ' 300x6< (FHDx=1920)
00,270                 YdimROW = 80 ' 80x12< (FHDy=1080)
00,271                 _TITLE "MASAKARI, revision 8.1+ 300x80, The 'Holy Axe' English Text Sidekick"
00,272             END IF
00,273         END IF
00,274     END IF
00,275 $ELSE
00,276     'IF _FILEEXISTS(PSPlike$ + "MxPlus_ToshibaTxL2_8x16.ttf") = 0 THEN
00,277     IF Mode4K = 0 THEN
00,278     IF wide& >= 1920 THEN
00,279         XdimCOL = 300 ' 300x6< (FHDx=1920)
00,280         YdimROW = 80 ' 80x12< (FHDy=1080)
00,281         _TITLE "MASAKARI, revision 8.1+ 300x80, The 'Holy Axe' English Text Sidekick"
00,282     END IF
00,283     END IF
00,284     'END IF
00,285 $END IF
00,286
00,287 '' SOUND PLAYER [
00,288 ''s& = _SNDOPEN(PSPlike$+"music_zapsplat_rapid_turnaround.mp3")
00,289 ''s& = _SNDOPEN(PSPlike$+"music_pioxonaq_agressive_lightning.mp3")
00,290 'IF s& <> 0 THEN
00,291 '    _SNDPLAY s& 'check for valid handle before using!
00,292 '    'The _SNDLOOP statement is like _SNDPLAY but the sound is looped. Uses a handle from the _SNDOPEN function.
00,293 '    'Example 1: Playing a previously opened sound at half volume.

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

00,294 '      '_SNDPLAYCOPY applause&, 0.5
00,295 '      volumeINT% = 9 '10 means 100% means 1.0, down to 0 i.e. 0..10 or 0,0.1,...1.0
00,296 '      StopFlag = 0
00,297 '      DO
00,298 '          k$ = INKEY$
00,299 '          SELECT CASE k$
00,300 '              CASE "n"
00,301 '                  IF volumeINT% >= 1 THEN volumeINT% = volumeINT% - 1
00,302 '                  _SNDVOL s&, volumeINT% / 10
00,303 '              CASE "m"
00,304 '                  IF volumeINT% <= 8 THEN volumeINT% = volumeINT% + 1 ' don't go 100% but up to 90%
00,305 '                  _SNDVOL s&, volumeINT% / 10
00,306 '              CASE "p"
00,307 '                  _SNDPAUSE s&
00,308 '              CASE " "
00,309 '                  _SNDPLAY s&
00,310 '              CASE CHR$(27)
00,311 '                  _SNDSTOP s&
00,312 '                  _SNDCLOSE s&
00,313 '                  StopFlag = 1
00,314 '          END SELECT
00,315 '          LOCATE 1, 1
00,316 '          IF StopFlag = 0 THEN PRINT "Volume: "; volumeINT%; " "; Position: "; INT(_SNDGETPOS(s&)); "second, up to"; INT(_SNDLEN(s&))
00,317 '      LOOP UNTIL StopFlag
00,318 'END IF
00,319 '' SOUND PLAYER ]
00,320
00,321 'spell [
00,322 DIM lfixed AS STRING * 32
00,323 TotalWrd& = 0
00,324 OPEN PSPluke$ + "masakari.ind" FOR BINARY AS #2
00,325 IF LOF(2) THEN
00,326     TotalWrd& = LOF(2) / 32
00,327     CLOSE #2
00,328 ELSE
00,329     CLOSE #2
00,330     KILL PSPluke$ + "masakari.ind"
00,331     OPEN PSPluke$ + "masakari.wrd" FOR BINARY AS #2
00,332     IF LOF(2) THEN
00,333         CLOSE #2
00,334         OPEN PSPluke$ + "masakari.wrd" FOR INPUT AS #1
00,335         OPEN PSPluke$ + "masakari.ind" FOR RANDOM ACCESS WRITE AS #2 LEN = 32
00,336         DO WHILE NOT EOF(1)
00,337             LINE INPUT #1, l$
00,338             lfixed$ = l$
00,339             PUT #2, , lfixed$: TotalWrd& = TotalWrd& + 1
00,340             FL2wrd& = 1
00,341         LOOP
00,342     CLOSE #1, #2

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,343 ELSE
00,344 CLOSE #2
00,345 KILL PSPluke$ + "masakari.wrd"
00,346 END IF
00,347 END IF
00,348 'spell ]
00,349
00,350 IF INSTR(LCASE$(COMMAND$), "/ascii") OR INSTR(LCASE$(COMMAND$), "-ascii") THEN
00,351 ELSE
00,352 GOTO SkipFontReviews
00,353 END IF
00,354 'SCREEN 0 'It is better to define your own mode, as:
00,355 handle& = _NEWIMAGE(98, 30, 0)
00,356 SCREEN handle&
00,357 _DEST handle&
00,358
00,359 ' _FONT _LOADFONT("C:\windows\fonts\cour.ttf", 32, "MONOSPACE")
00,360 ' _FONT _LOADFONT("C:\windows\fonts\lucon.ttf", 32, "MONOSPACE")
00,361 ' _FONT _LOADFONT("C:\windows\fonts\MxPlus_Cordata_PPC-400.ttf", 32, "MONOSPACE")
00,362 ' _FONT _LOADFONT("C:\windows\fonts\MxPlus_ToshibaTxl2_8x16.ttf", 32, "MONOSPACE")
00,363 ' _FONT _LOADFONT(PSPluke$ + "MxPlus_ToshibaTxl2_8x16.ttf", 32, "MONOSPACE")
00,364
00,365 'RESTORE Microsoft_pc_cp437 'United States MS DOS
00,366 'RESTORE Microsoft_windows_cp1250 'WINDOWS in Central European and Eastern European languages that use Latin script, such as Polish, Czech, Slovak, Hungarian, Slovene, Bosnian, Croatian, Serbian (Latin script), Romanian
and Albanian. It may also be used with the German language.
00,367 'RESTORE Microsoft_windows_cp1251 'Cyrillic alphabet such as Russian, Bulgarian, Serbian Cyrillic and other languages. It is the most widely used for encoding the Bulgarian, Serbian and Macedonian languages.
00,368 'RESTORE Microsoft_pc_cpMIK 'Cyrillic Bulgarian Pravetz 16 for MS-DOS
00,369 COLOR 4
00,370 IF INSTR(LCASE$(COMMAND$), "/ascii_gesch") OR INSTR(LCASE$(COMMAND$), "-ascii_gesch") THEN
00,371 RESTORE Microsoft_pc_cpGESCH 'My codepage a.k.a. Gesch
00,372 PRINT "Showing GRAPHEMES of Gesch (a.k.a. Schpitz) codepage,": PRINT "it allows browsing German/Italian/French/Spanish/Bulgarian (and its dialect Russian :P):"
00,373 ELSE
00,374 RESTORE Microsoft_windows_cp1252
00,375 PRINT "Showing GRAPHEMES of cp1252 codepage - Windows Western languages with Latin alphabet,":
00,376 PRINT "it allows browsing... ugh, it sucks:"
00,377 END IF
00,378
00,379 IF INSTR(LCASE$(COMMAND$), "/ascii_gesch") OR INSTR(LCASE$(COMMAND$), "-ascii_gesch") THEN
00,380 FOR ASCIIcode = 0 TO 6 'German A:a:O:o:U:u: ss
00,381 READ unicode
00,382 _MAPUNICODE unicode TO ASCIIcode
00,383 NEXT
00,384 FOR ASCIIcode = 7 TO 7 'Integral high
00,385 READ unicode
00,386 _MAPUNICODE unicode TO ASCIIcode
00,387 NEXT
00,388 FOR ASCIIcode = 8 TO 8 'Integral low
00,389 READ unicode
00,390 _MAPUNICODE unicode TO ASCIIcode

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,391 NEXT
00,392 FOR ASCIIcode = 14 TO 15 'French C,c,
00,393 READ unicode
00,394 _MAPUNICODE unicode TO ASCIIcode
00,395 NEXT
00,396 FOR ASCIIcode = 16 TO 27 'French/Italian E'e' A'a'E'e'I'i'O'o'U'u'
00,397 READ unicode
00,398 _MAPUNICODE unicode TO ASCIIcode
00,399 NEXT
00,400 'why 28..31 are not available?!
00,401 FOR ASCIIcode = 128 TO 255
00,402 READ unicode
00,403 _MAPUNICODE unicode TO ASCIIcode
00,404 NEXT
00,405 FOR ASCIIcode = 220 TO 223 ' the unnecessary 4 bold drawing chars are replaced with French Æ ae I: i:
00,406 READ unicode
00,407 _MAPUNICODE unicode TO ASCIIcode
00,408 NEXT
00,409 FOR ASCIIcode = 127 TO 127 'almost equal to
00,410 READ unicode
00,411 _MAPUNICODE unicode TO ASCIIcode
00,412 NEXT
00,413 FOR ASCIIcode = 181 TO 190 'Cyrillic short vowels
00,414 READ unicode
00,415 _MAPUNICODE unicode TO ASCIIcode
00,416 NEXT
00,417 FOR ASCIIcode = 198 TO 207 'Cyrillic short vowels
00,418 READ unicode
00,419 _MAPUNICODE unicode TO ASCIIcode
00,420 NEXT
00,421 FOR ASCIIcode = 208 TO 216 'misc, Spanish
00,422 READ unicode
00,423 _MAPUNICODE unicode TO ASCIIcode
00,424 NEXT
00,425 FOR ASCIIcode = 28 TO 31 'draw animated block
00,426 READ unicode
00,427 _MAPUNICODE unicode TO ASCIIcode
00,428 NEXT
00,429 FOR ASCIIcode = 11 TO 12 'draw animated block
00,430 READ unicode
00,431 _MAPUNICODE unicode TO ASCIIcode
00,432 NEXT
00,433 END IF
00,434
00,435 toggleCyanPurpleWhite = 3
00,436 COLOR toggleCyanPurpleWhite
00,437 IF cnt MOD 16 = 0 THEN PRINT: PRINT STRING$(3 - LEN(LTRIM$(STR$(cnt))), "0") + LTRIM$(STR$(cnt)); " ";
00,438 FOR i = 0 TO 255
00,439 'IF i <> 11 AND i <> 12 AND i <> 9 AND i <> 7 AND i <> 13 AND i <> 10 THEN

```

```

00,440 PRINT CHR$(i) + " ";
00,441 'ELSE
00,442 'PRINT " ";
00,443 'END IF
00,444 cnt = cnt + 1
00,445 IF (cnt <= 255) AND (cnt MOD 16 = 0) THEN
00,446 IF toggleCyanPurpleWhite = 3 THEN
00,447 toggleCyanPurpleWhite = 7
00,448 ELSEIF toggleCyanPurpleWhite = 7 THEN
00,449 toggleCyanPurpleWhite = 9
00,450 ELSE
00,451 toggleCyanPurpleWhite = 3
00,452 END IF
00,453 COLOR toggleCyanPurpleWhite
00,454 PRINT: PRINT STRING$(3 - LEN(LTRIM$(STR$(cnt))), "0") + LTRIM$(STR$(cnt)); " ";
00,455 END IF
00,456 NEXT
00,457 COLOR 4
00,458 IF INSTR(LCASE$(COMMAND$), "/ascii_gesch") OR INSTR(LCASE$(COMMAND$), "-ascii_gesch") THEN
00,459 PRINT: PRINT: PRINT "Example (forming a long integral from 209 179 210): ": PRINT CHR$(209); " Useful for": PRINT CHR$(179); " making paragraphs": PRINT CHR$(210); " of your own."
00,460 ELSE
00,461 PRINT: PRINT: PRINT "Example (forming a long integral from 244 179 245): ": PRINT CHR$(244); " Useful for": PRINT CHR$(179); " making paragraphs": PRINT CHR$(245); " of your own."
00,462 END IF
00,463 COLOR 7
00,464 END
00,465
00,466 Microsoft_pc_cp737: 'Greek MS DOS displays Greek alphabet for algebraic formulas.
00,467 DATA 913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928
00,468 DATA 929,931,932,933,934,935,936,937,945,946,947,948,949,950,951,952
00,469 DATA 953,954,955,956,957,958,959,960,961,963,962,964,965,966,967,968
00,470 DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488
00,471 DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575
00,472 DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600
00,473 DATA 969,940,941,942,970,943,972,973,971,974,902,904,905,906,908,910
00,474 DATA 911,177,8805,8804,938,939,247,8776,176,8729,183,8730,8319,178,9632,160
00,475
00,476 Microsoft_pc_cp775: 'Estonian, Lithuanian and Latvian languages.
00,477 DATA 262,252,233,257,228,291,229,263,322,275,342,343,299,377,196,197
00,478 DATA 201,230,198,333,246,290,162,346,347,214,220,248,163,216,215,164
00,479 DATA 256,298,243,379,380,378,8221,166,169,174,172,189,188,321,171,187
00,480 DATA 9617,9618,9619,9474,9508,260,268,280,278,9571,9553,9559,9565,302,352,9488
00,481 DATA 9492,9524,9516,9500,9472,9532,370,362,9562,9556,9577,9574,9568,9552,9580,381
00,482 DATA 261,269,281,279,303,353,371,363,382,9496,9484,9608,9604,9612,9616,9600
00,483 DATA 211,223,332,323,245,213,181,324,310,311,315,316,326,274,325,8217
00,484 DATA 173,177,8220,190,182,167,247,8222,176,8729,183,185,179,178,9632,160
00,485
00,486 Microsoft_pc_cp850: 'Western Europe, Spain, England
00,487 DATA 199,252,233,226,228,224,229,231,234,235,232,239,238,236,196,197
00,488 DATA 201,230,198,244,246,242,251,249,255,214,220,248,163,216,215,402

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```
00,489 DATA 225,237,243,250,241,209,170,186,191,174,172,189,188,161,171,187
00,490 DATA 9617,9618,9619,9474,9508,193,194,192,169,9571,9553,9559,9565,162,165,9488
00,491 DATA 9492,9524,9516,9500,9472,9532,227,195,9562,9556,9577,9574,9568,9552,9580,164
00,492 DATA 240,208,202,203,200,305,205,206,207,9496,9484,9608,9604,166,204,9600
00,493 DATA 211,223,212,210,245,213,181,254,222,218,219,217,253,221,175,180
00,494 DATA 173,177,8215,190,182,167,247,184,176,168,183,185,179,178,9632,160
00,495
00,496 Microsoft_pc_cp852: 'Central European languages that use Latin script such as Bosnian, Croatian, Czech, Hungarian, Polish, Romanian, Serbian or Slovak.
00,497 DATA 199,252,233,226,228,367,263,231,322,235,336,337,238,377,196,262
00,498 DATA 201,313,314,244,246,317,318,346,347,214,220,356,357,321,215,269
00,499 DATA 225,237,243,250,260,261,381,382,280,281,172,378,268,351,171,187
00,500 DATA 9617,9618,9619,9474,9508,193,194,282,350,9571,9553,9559,9565,379,380,9488
00,501 DATA 9492,9524,9516,9500,9472,9532,258,259,9562,9556,9577,9574,9568,9552,9580,164
00,502 DATA 273,272,270,203,271,327,205,206,283,9496,9484,9608,9604,354,366,9600
00,503 DATA 211,223,212,323,324,328,352,353,340,218,341,368,253,221,355,180
00,504 DATA 173,733,731,711,728,167,247,184,176,168,729,369,344,345,9632,160
00,505
00,506 Microsoft_pc_cp855: 'Cyrillic code page to be used under MS-DOS
00,507 DATA 1106,1026,1107,1027,1105,1025,1108,1028,1109,1029,1110,1030,1111,1031,1112,1032
00,508 DATA 1113,1033,1114,1034,1115,1035,1116,1036,1118,1038,1119,1039,1102,1070,1098,1066
00,509 DATA 1072,1040,1073,1041,1094,1062,1076,1044,1077,1045,1092,1060,1075,1043,171,187
00,510 DATA 9617,9618,9619,9474,9508,1093,1061,1080,1048,9571,9553,9559,9565,1081,1049,9488
00,511 DATA 9492,9524,9516,9500,9472,9532,1082,1050,9562,9556,9577,9574,9568,9552,9580,164
00,512 DATA 1083,1051,1084,1052,1085,1053,1086,1054,1087,9496,9484,9608,9604,1055,1103,9600
00,513 DATA 1071,1088,1056,1089,1057,1090,1058,1091,1059,1078,1046,1074,1042,1100,1068,8470
00,514 DATA 173,1099,1067,1079,1047,1096,1064,1101,1069,1097,1065,1095,1063,167,9632,160
00,515
00,516 Microsoft_pc_cp857: 'Turkish MS DOS
00,517 DATA 199,252,233,226,228,224,229,231,234,235,232,239,238,305,196,197
00,518 DATA 201,230,198,244,246,242,251,249,304,214,220,248,163,216,350,351
00,519 DATA 225,237,243,250,241,209,286,287,191,174,172,189,188,161,171,187
00,520 DATA 9617,9618,9619,9474,9508,193,194,192,169,9571,9553,9559,9565,162,165,9488
00,521 DATA 9492,9524,9516,9500,9472,9532,227,195,9562,9556,9577,9574,9568,9552,9580,164
00,522 DATA 186,170,202,203,200,0,205,206,207,9496,9484,9608,9604,166,204,9600
00,523 DATA 211,223,212,210,245,213,181,0,215,218,219,217,236,255,175,180
00,524 DATA 173,177,0,190,182,167,247,184,176,168,183,185,179,178,9632,160
00,525
00,526 Microsoft_pc_cp860: 'Portuguese language. MS DOS
00,527 DATA 199,252,233,226,227,224,193,231,234,202,232,205,212,236,195,194
00,528 DATA 201,192,200,244,245,242,218,249,204,213,220,162,163,217,8359,211
00,529 DATA 225,237,243,250,241,209,170,186,191,210,172,189,188,161,171,187
00,530 DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488
00,531 DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575
00,532 DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600
00,533 DATA 945,223,915,960,931,963,181,964,934,920,937,948,8734,966,949,8745
00,534 DATA 8801,177,8805,8804,8992,8993,247,8776,176,8729,183,8730,8319,178,9632,160
00,535
00,536 Microsoft_pc_cp861: 'Icelandic language (as well as other Nordic languages). MS DOS
00,537 DATA 199,252,233,226,228,224,229,231,234,235,232,208,240,222,196,197
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```
00,538 DATA 201,230,198,244,246,254,251,221,253,214,220,248,163,216,8359,402
00,539 DATA 225,237,243,250,193,205,211,218,191,8976,172,189,188,161,171,187
00,540 DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488
00,541 DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575
00,542 DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600
00,543 DATA 945,223,915,960,931,963,181,964,934,920,937,948,8734,966,949,8745
00,544 DATA 8801,177,8805,8804,8992,8993,247,8776,176,8729,183,8730,8319,178,9632,160
00,545
00,546 Microsoft_pc_cp862: 'Hebrew letters in positions 809A hex, but otherwise it is identical to CP437. Now obsolete, see CP1255
00,547 DATA 1488,1489,1490,1491,1492,1493,1494,1495,1496,1497,1498,1499,1500,1501,1502,1503
00,548 DATA 1504,1505,1506,1507,1508,1509,1510,1511,1512,1513,1514,162,163,165,8359,402
00,549 DATA 225,237,243,250,241,209,170,186,191,8976,172,189,188,161,171,187
00,550 DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488
00,551 DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575
00,552 DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600
00,553 DATA 945,223,915,960,931,963,181,964,934,920,937,948,8734,966,949,8745
00,554 DATA 8801,177,8805,8804,8992,8993,247,8776,176,8729,183,8730,8319,178,9632,160
00,555
00,556 Microsoft_pc_cp863: 'French language (mainly in Canada). MS DOS
00,557 DATA 199,252,233,226,194,224,182,231,234,235,232,239,238,8215,192,167
00,558 DATA 201,200,202,244,203,207,251,249,164,212,220,162,163,217,219,402
00,559 DATA 166,180,243,250,168,184,179,175,206,8976,172,189,188,190,171,187
00,560 DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488
00,561 DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575
00,562 DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600
00,563 DATA 945,223,915,960,931,963,181,964,934,920,937,948,8734,966,949,8745
00,564 DATA 8801,177,8805,8804,8992,8993,247,8776,176,8729,183,8730,8319,178,9632,160
00,565
00,566 Microsoft_pc_cp864: 'Arabic MS DOS
00,567 DATA 176,183,8729,8730,9618,9472,9474,9532,9508,9516,9500,9524,9488,9484,9492,9496
00,568 DATA 946,8734,966,177,189,188,8776,171,187,65271,65272,0,0,65275,65276,0
00,569 DATA 160,173,65154,163,164,65156,0,0,65166,65167,65173,65177,1548,65181,65185,65189
00,570 DATA 1632,1633,1634,1635,1636,1637,1638,1639,1640,1641,65233,1563,65201,65205,65209,1567
00,571 DATA 162,65152,65153,65155,65157,65226,65163,65165,65169,65171,65175,65179,65183,65187,65191,65193
00,572 DATA 65195,65197,65199,65203,65207,65211,65215,65217,65221,65227,65231,166,172,247,215,65225
00,573 DATA 1600,65235,65239,65243,65247,65251,65255,65259,65261,65263,65267,65213,65228,65230,65229,65249
00,574 DATA 65149,1617,65253,65257,65260,65264,65266,65232,65237,65269,65270,65245,65241,65265,9632,0
00,575
00,576 Microsoft_pc_cp865: 'Nordic languages (except Icelandic, for which CP861 is used). MS DOS
00,577 DATA 199,252,233,226,228,224,229,231,234,235,232,239,238,236,196,197
00,578 DATA 201,230,198,244,246,242,251,249,255,214,220,248,163,216,8359,402
00,579 DATA 225,237,243,250,241,209,170,186,191,8976,172,189,188,161,171,164
00,580 DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488
00,581 DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575
00,582 DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600
00,583 DATA 945,223,915,960,931,963,181,964,934,920,937,948,8734,966,949,8745
00,584 DATA 8801,177,8805,8804,8992,8993,247,8776,176,8729,183,8730,8319,178,9632,160
00,585
00,586 Microsoft_pc_cp866: 'Cyrillic alphabetical order code page to be used with MS-DOS
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

00,635
00,636 Microsoft_windows_cp1253: 'Greek (but not polytonic Greek) Not fully compatible with ISO 8859-7 (? is located differently).
00,637 DATA 8364,0,8218,402,8222,8230,8224,8225,0,8240,0,8249,0,0,0,0
00,638 DATA 0,8216,8217,8220,8221,8226,8211,8212,0,8482,0,8250,0,0,0,0
00,639 DATA 160,901,902,163,164,165,166,167,168,169,0,171,172,173,174,8213
00,640 DATA 176,177,178,179,900,181,182,183,904,905,906,187,908,189,910,911
00,641 DATA 912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927
00,642 DATA 928,929,0,931,932,933,934,935,936,937,938,939,940,941,942,943
00,643 DATA 944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959
00,644 DATA 960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,0
00,645
00,646 Microsoft_windows_cp1254: 'Turkish
00,647 DATA 8364,0,8218,402,8222,8230,8224,8225,710,8240,352,8249,338,0,0,0
00,648 DATA 0,8216,8217,8220,8221,8226,8211,8212,732,8482,353,8250,339,0,0,376
00,649 DATA 160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175
00,650 DATA 176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191
00,651 DATA 192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207
00,652 DATA 286,209,210,211,212,213,214,215,216,217,218,219,220,304,350,223
00,653 DATA 224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239
00,654 DATA 287,241,242,243,244,245,246,247,248,249,250,251,252,305,351,255
00,655
00,656 Microsoft_windows_cp1255: 'Hebrew Windows. Modern applications prefer [https://en.wikipedia.org/wiki/UTF-8 UTF-8] or UTF-16 http://www.fileformat.info/info/charset/UTF-16/list.htm to Windows 1255.
00,657 DATA 8364,0,8218,402,8222,8230,8224,8225,710,8240,0,8249,0,0,0,0
00,658 DATA 0,8216,8217,8220,8221,8226,8211,8212,732,8482,0,8250,0,0,0,0
00,659 DATA 160,161,162,163,8362,165,166,167,168,169,215,171,172,173,174,175
00,660 DATA 176,177,178,179,180,181,182,183,184,185,247,187,188,189,190,191
00,661 DATA 1456,1457,1458,1459,1460,1461,1462,1463,1464,1465,0,1467,1468,1469,1470,1471
00,662 DATA 1472,1473,1474,1475,1520,1521,1522,1523,1524,0,0,0,0,0,0
00,663 DATA 1488,1489,1490,1491,1492,1493,1494,1495,1496,1497,1498,1499,1500,1501,1502,1503
00,664 DATA 1504,1505,1506,1507,1508,1509,1510,1511,1512,1513,1514,0,0,8206,8207,0
00,665
00,666 Microsoft_windows_cp1256: 'Arabic Latin Windows
00,667 DATA 8364,1662,8218,402,8222,8230,8224,8225,710,8240,1657,8249,338,1670,1688,1672
00,668 DATA 1711,8216,8217,8220,8221,8226,8211,8212,1705,8482,1681,8250,339,8204,8205,1722
00,669 DATA 160,1548,162,163,164,165,166,167,168,169,1726,171,172,173,174,175
00,670 DATA 176,177,178,179,180,181,182,183,184,185,1563,187,188,189,190,1567
00,671 DATA 1729,1569,1570,1571,1572,1573,1574,1575,1576,1577,1578,1579,1580,1581,1582,1583
00,672 DATA 1584,1585,1586,1587,1588,1589,1590,215,1591,1592,1593,1594,1600,1601,1602,1603
00,673 DATA 224,1604,226,1605,1606,1607,1608,231,232,233,234,235,1609,1610,238,239
00,674 DATA 1611,1612,1613,1614,244,1615,1616,247,1617,249,1618,251,252,8206,8207,1746
00,675
00,676 Microsoft_windows_cp1257: 'Estonian (although that can also be written with Windows-1252), Latvian and Lithuanian languages under Microsoft Windows. It is also possible to write Polish and German.
00,677 DATA 8364,0,8218,0,8222,8230,8224,8225,0,8240,0,8249,0,168,711,184
00,678 DATA 0,8216,8217,8220,8221,8226,8211,8212,0,8482,0,8250,0,175,731,0
00,679 DATA 160,0,162,163,164,0,166,167,216,169,342,171,172,173,174,198
00,680 DATA 176,177,178,179,180,181,182,183,248,185,343,187,188,189,190,230
00,681 DATA 260,302,256,262,196,197,280,274,268,201,377,278,290,310,298,315
00,682 DATA 352,323,325,211,332,213,214,215,370,321,346,362,220,379,381,223
00,683 DATA 261,303,257,263,228,229,281,275,269,233,378,279,291,311,299,316

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

00,684 DATA 353,324,326,243,333,245,246,247,371,322,347,363,252,380,382,729

00,685

00,686 Microsoft_windows_cp1258: 'Vietnamese. [https://en.wikipedia.org/wiki/UTF-8 UTF-8] is the preferred encoding for Vietnamese in modern applications.

00,687 DATA 8364,0,8218,402,8222,8230,8224,8225,710,8240,0,8249,338,0,0,0

00,688 DATA 0,8216,8217,8220,8221,8226,8211,8212,732,8482,0,8250,339,0,0,376

00,689 DATA 160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175

00,690 DATA 176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191

00,691 DATA 192,193,194,258,196,197,198,199,200,201,202,203,768,205,206,207

00,692 DATA 272,209,777,211,212,416,214,215,216,217,218,219,220,431,771,223

00,693 DATA 224,225,226,259,228,229,230,231,232,233,234,235,769,237,238,239

00,694 DATA 273,241,803,243,244,417,246,247,248,249,250,251,252,432,8363,255

00,695

00,696 Microsoft_pc_cp437: 'United States MS DOS

00,697 DATA 199,252,233,226,228,224,229,231,234,235,232,239,238,236,196,197

00,698 DATA 201,230,198,244,246,242,251,249,255,214,220,162,163,165,8359,402

00,699 DATA 225,237,243,250,241,209,170,186,191,8976,172,189,188,161,171,187

00,700 DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488

00,701 DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575

00,702 DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600

00,703 DATA 945,223,915,960,931,963,181,964,934,920,937,948,8734,966,949,8745

00,704 DATA 8801,177,8805,8804,8992,8993,247,8776,176,8729,183,8730,8319,178,9632,160

00,705

00,706 Microsoft_windows_cp1251: 'Cyrillic alphabet such as Russian, Bulgarian, Serbian Cyrillic and other languages. It is the most widely used for encoding the Bulgarian, Serbian and Macedonian languages.

00,707 DATA 1026,1027,8218,1107,8222,8230,8224,8225,8364,8240,1033,8249,1034,1036,1035,1039

00,708 DATA 1106,8216,8217,8220,8221,8226,8211,8212,0,8482,1113,8250,1114,1116,1115,1119

00,709 DATA 160,1038,1118,1032,164,1168,166,167,1025,169,1028,171,172,173,174,1031

00,710 DATA 176,177,1030,1110,1169,181,182,183,1105,8470,1108,187,1112,1029,1109,1111

00,711 DATA 1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055

00,712 DATA 1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071

00,713 DATA 1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087

00,714 DATA 1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103

00,715

00,716 Microsoft_pc_cpMIK: 'Cyrillic Bulgarian Pravetz 16 for MS-DOS

00,717 DATA 1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055

00,718 DATA 1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071

00,719 DATA 1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087

00,720 DATA 1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103

00,721 DATA 9492,9524,9516,9500,9472,9532,9571,9553,9562,9566,9577,9574,9568,9552,9580,9488

00,722 DATA 9617,9618,9619,9474,9508,8470,167,9559,9565,9496,9484,9608,9604,9612,9616,9600

00,723 DATA 945,223,915,960,931,963,181,964,934,920,937,948,8734,966,949,8745

00,724 DATA 8801,177,8805,8804,8992,8993,247,8776,176,8729,183,8730,8319,178,9632,160

00,725

00,726 Microsoft_pc_cpGESCH: 'Schpitz/Gesch (a.k.a. Georgievica a.k.a. Geschovica) is Sanmayce's layout, combining the MIK and 437, in this way: (NOT RECODING only 009,010,013):

00,727 ' shpitz

00,728

00,729 ' DEFINITIONS

00,730 ' top or extreme part

00,731 ' the ultimate, the best of, "cool"

00,732 ' typical, classic, a perfect example of; definitively

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

00,733
00,734 ' LANGUAGES OF ORIGIN
00,735 ' Yiddish
00,736
00,737 ' ETYMOLOGY
00,738 ' ????? shpits 'tip, peak'
00,739
00,740 ' ALTERNATIVE SPELLINGS
00,741 ' schpitz, shpits, shpitz, shpitsy
00,742
00,743 'NOTES
00,744 'Steinmetz, "Yiddish and English" has an entry for shpits, but the meaning is simply 'tip.'
00,745
00,746 ' ASCII 000..031 have to accomodate German and French 'H203e = 8254 is upperscore 'H00DC=220
00,747 DATA 196,228,214,246,&H00DC,&H00FC,223,&H00c1,&H00e1,&h00C7,&H00E7
00,748 'A:a:O:o:U:u:ss A' a' C'c'
00,749 DATA &H0c9,&H00e9,&H0c0,&H00e0,&H00c8,&H00e8,&H00cc,&H00ec,&H00d2,&H00f2,&H00d9,&H00f9
00,750 ' E'e' A'a' E'e' I'i' O'o' U'u'
00,751
00,752 'DATA &H00cb,&H00eb,&H00cf,&H00ef
00,753 ' E:e' I:i' E:e' are present in Russian, so remove them
00,754
00,755 'First half of big Cyrillic letters:
00,756 DATA 1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055
00,757 'Second half of big Cyrillic letters:
00,758 DATA 1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071
00,759 'First half of small Cyrillic letters:
00,760 DATA 1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087
00,761 '1st third of CP437 drawing symbols:
00,762 DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488
00,763 '2nd third of CP437 drawing symbols:
00,764 DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575
00,765 '3rd third of CP437 drawing symbols:
00,766 DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600
00,767 'Second half of small Cyrillic letters:
00,768 DATA 1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103
00,769 'Last 16 of CP437 symbols:
00,770 '
00,771 'DATA 8216,8217,8218,8219,8220,8221,8222,8223,176,1118,1025,1105,171,187,175,8230
00,772 '
00,773 DATA 8216,8217,8218,8219,8220,8221,8222,8223,176,&h00b5,1025,1105,171,187,175,8230
00,774
00,775 DATA &H00c6,&H00E6,&H0152,&H0153
00,776 ' AE ae OEoe
00,777
00,778 DATA &h2248
00,779 'almost equal to
00,780 'DATA &H221a
00,781 'square root

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

00,782
00,783 DATA &H0102,&H0103,&H0114,&H0115,&H014e,&H014f,&H040e,&H045e,&H00cf,&H00ef
00,784 'Cyrillic short vowels: AaEeOoYy I:i:
00,785 DATA &H00c2,&H00e2,&H00ca,&H00ea,&H00ce,&H00ee,&H00d4,&H00f4,&H00db,&H00fb
00,786 ' A^a^ E^e^ I^i^ O^o^ U^u^
00,787
00,788 DATA &h2017,&h2320,&h2321,&h00a1,&h00bf,&h00d1,&h00f1,&h00dd,&h00fd
00,789 'doubleunderline IntegralH Integrall, Spanish: r! r? N~ n~ Y` y`
00,790
00,791 DATA &h00cd,&h00ed,&h00da,&h00fa
00,792 ' I`i`U`u`
00,793
00,794 DATA &h00D3,&h00F3
00,795 ' O` o`
00,796
00,797 ' Portuguese makes use of five diacritics: the cedilla (ç), acute accent (á é í ó ú), circumflex accent (â, ê, ô), tilde (ã, õ), and grave accent (à, and rarely è, ì, ò, and ù).
00,798
00,799 SkipFontReviews:
00,800 handle& = _NEWIMAGE(XdimCOL + IndigoField, YdimROW + 1, 0)
00,801 SCREEN handle&
00,802 _DEST handle&
00,803
00,804 ' _DEST 0 refers to the present program SCREEN. You can use 0 to refer to the present program
00,805 ' SCREEN.
00,806
00,807 ' _FONT 16 'wish we could use the old 8x16 bitmap/raster fonts from DOS times...
00,808
00,809 'Let's think for people using 4K, then 32px is to be used. Now, 128col x 8 = 1024 (fits well in 1680x) or 198col x 8 = 1584 (fits well in FHD)
00,810 'IF _FILEEXISTS("C:\windows\fonts\MxPlus_ToshibaTxL2_8x16.ttf") = 0 THEN
00,811 $IF BIGORSMALL = 0 THEN
00,812     IF Mode4K = 0 THEN
00,813         ' _FONT _LOADFONT(PSPlike$ + "cour.ttf", 14, "MONOSPACE") ' 8x14 is not bad at all!
00,814         ' _FONT _LOADFONT(PSPlike$ + "cour.ttf", 16, "MONOSPACE")
00,815         ' _FONT _LOADFONT(PSPlike$ + "lucon.ttf", 14, "MONOSPACE") ' 8x14 is not bad at all!
00,816         ' _FONT _LOADFONT(PSPlike$ + "lucon.ttf", 16, "MONOSPACE")
00,817         'If not a single line is uncommented, above, then the EMULATED (Trident?) font is in use.
00,818         _FONT _LOADFONT(PSPlike$ + "Px437_DOS-V_re_JPN12.ttf", 12, "MONOSPACE") '2021-Apr-23, use it on old laptops with 1680- by x, it is 6x12
00,819         ' _FONT _LOADFONT(PSPlike$ + "lucon.ttf", 12, "MONOSPACE") '2021-Apr-23, use it on old laptops with 1680- by x, it is 6x12
00,820     ELSE
00,821         ' _FONT _LOADFONT(PSPlike$ + "cour.ttf", 32, "MONOSPACE")
00,822         _FONT _LOADFONT(PSPlike$ + "lucon.ttf", 32, "MONOSPACE")
00,823         'If not a single line is uncommented, above, then the EMULATED (Trident?) font is in use.
00,824     END IF
00,825 $ELSE
00,826     IF _FILEEXISTS(PSPlike$ + "MxPlus_ToshibaTxL2_8x16.ttf") = 0 THEN
00,827         IF Mode4K = 0 THEN
00,828             ' _FONT _LOADFONT(PSPlike$ + "cour.ttf", 14, "MONOSPACE") ' 8x14 is not bad at all!
00,829             ' _FONT _LOADFONT(PSPlike$ + "cour.ttf", 16, "MONOSPACE")
00,830             ' _FONT _LOADFONT(PSPlike$ + "lucon.ttf", 14, "MONOSPACE") ' 8x14 is not bad at all!

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

00,831      '_FONT_LOADFONT(PSPlike$ + "lucon.ttf", 16, "MONOSPACE")
00,832      'If not a single line is uncommented, above, then the EMULATED (Trident?) font is in use.
00,833      _FONT_LOADFONT(PSPlike$ + "Px437_DOS-V_re_JPN12.ttf", 12, "MONOSPACE") '2021-Apr-23, use it on old laptops with 1680- by x, it is 6x12
00,834      '_FONT_LOADFONT(PSPlike$ + "lucon.ttf", 12, "MONOSPACE") '2021-Apr-23, use it on old laptops with 1680- by x, it is 6x12
00,835      ELSE
00,836      '_FONT_LOADFONT(PSPlike$ + "cour.ttf", 32, "MONOSPACE")
00,837      _FONT_LOADFONT(PSPlike$ + "lucon.ttf", 32, "MONOSPACE")
00,838      'If not a single line is uncommented, above, then the EMULATED (Trident?) font is in use.
00,839      END IF
00,840      ELSE
00,841      IF Mode4K = 0 THEN
00,842      IF FONT4x = 1 THEN
00,843      _FONT_LOADFONT(PSPlike$ + "MxPlus_ToshibaTxL2_8x16.ttf", 32, "MONOSPACE") 'Toshiba rules...
00,844      ELSE
00,845      _FONT_LOADFONT(PSPlike$ + "MxPlus_ToshibaTxL2_8x16.ttf", 16, "MONOSPACE") 'Toshiba rules...
00,846      END IF
00,847      '_FONT_LOADFONT(PSPlike$ + "MxPlus_Cordata_PPC-400.ttf", 16, "MONOSPACE") 'Beautiful, yet the '2' has a missing dot - up-right
00,848      ELSE
00,849      _FONT_LOADFONT(PSPlike$ + "MxPlus_ToshibaTxL2_8x16.ttf", 32, "MONOSPACE")
00,850      '_FONT_LOADFONT(PSPlike$ + "MxPlus_Cordata_PPC-400.ttf", 32, "MONOSPACE")
00,851      END IF
00,852      END IF
00,853 $END IF
00,854
00,855 'RESTORE Microsoft_windows_cp1252
00,856 RESTORE Microsoft_pc_cpGESCH 'My codepage a.k.a. Gesch
00,857 'RESTORE Microsoft_windows_cp1251
00,858
00,859 FOR ASCIIcode = 0 TO 6 'German A:a:O:o:U:u: ss
00,860     READ unicode
00,861     _MAPUNICODE unicode TO ASCIIcode
00,862 NEXT
00,863 FOR ASCIIcode = 7 TO 7 'Integral high
00,864     READ unicode
00,865     _MAPUNICODE unicode TO ASCIIcode
00,866 NEXT
00,867 FOR ASCIIcode = 8 TO 8 'Integral low
00,868     READ unicode
00,869     _MAPUNICODE unicode TO ASCIIcode
00,870 NEXT
00,871 FOR ASCIIcode = 14 TO 15 'French C,c,
00,872     READ unicode
00,873     _MAPUNICODE unicode TO ASCIIcode
00,874 NEXT
00,875 FOR ASCIIcode = 16 TO 27 'French/Italian E'e' A'a'E'e'I'i'O'o'U'u'
00,876     READ unicode
00,877     _MAPUNICODE unicode TO ASCIIcode
00,878 NEXT
00,879 'why 28..31 are not available?'
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

00,880 FOR AscIICode = 128 TO 255
00,881   READ unicode
00,882   _MAPUNICODE unicode TO AscIICode
00,883 NEXT
00,884 FOR AscIICode = 220 TO 223 ' the unnecessary 4 bold drawing chars are replaced with French AE ae I: i:
00,885   READ unicode
00,886   _MAPUNICODE unicode TO AscIICode
00,887 NEXT
00,888 FOR AscIICode = 127 TO 127 'almost equal to
00,889   READ unicode
00,890   _MAPUNICODE unicode TO AscIICode
00,891 NEXT
00,892 FOR AscIICode = 181 TO 190 'Cyrillic short vowels
00,893   READ unicode
00,894   _MAPUNICODE unicode TO AscIICode
00,895 NEXT
00,896 FOR AscIICode = 198 TO 207 'Cyrillic short vowels
00,897   READ unicode
00,898   _MAPUNICODE unicode TO AscIICode
00,899 NEXT
00,900 FOR AscIICode = 208 TO 216 'misc, Spanish
00,901   READ unicode
00,902   _MAPUNICODE unicode TO AscIICode
00,903 NEXT
00,904 FOR AscIICode = 28 TO 31 'draw animated block
00,905   READ unicode
00,906   _MAPUNICODE unicode TO AscIICode
00,907 NEXT
00,908 FOR AscIICode = 11 TO 12 'draw animated block
00,909   READ unicode
00,910   _MAPUNICODE unicode TO AscIICode
00,911 NEXT
00,912
00,913 ' Either one must be uncommented:
00,914 _DISPLAY
00,915 '_AUTODISPLAY 'no need of refreshing
00,916
00,917 PurpleFlag = 0
00,918
00,919 NormalFRGr = 3
00,920 IF PurpleFlag THEN NormalFRGr = 9
00,921 NormalBCKGr = 0
00,922 IF PurpleFlag THEN NormalBCKGr = 0
00,923
00,924 InverseFRGr = 0
00,925 IF PurpleFlag THEN InverseFRGr = 8
00,926 InverseBCKGr = 3
00,927 IF PurpleFlag THEN InverseBCKGr = 0
00,928

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```
00,929 COLOR NormalFRGr, NormalBCKGr
00,930
00,931 ShowASCIIart
00,932
00,933 $IF WINDOWS THEN
00,934     i = 0
00,935     DO
00,936         i = i + 1
00,937         setting$ = ENVIRON$(i) ' get a setting from the list
00,938         IF INSTR(setting$, "NUMBER_OF_PROCESSORS") THEN NUMBER_OF_PROCESSORS$ = setting$
00,939     LOOP UNTIL setting$ = ""
00,940     PRINT NUMBER_OF_PROCESSORS$
00,941
00,942     ' For some reason the Drag-and-Drop stops working when running as Administrator?! [
00,943     'C$ = ENVIRON$("ALLUSERSPROFILE") 'try desktop for all users
00,944     'SHORTCUT$ = C$ + "\Desktop\" + "Masakari.log.txt" 'create filename for the desktop
00,945     ' For some reason the Drag-and-Drop stops working when running as Administrator?! ]
00,946     SHORTCUT$ = _STARTDIR$ + "\" + "Masakari.log.txt"
00,947     OPEN SHORTCUT$ FOR APPEND AS #13
00,948     PRINT #13,
00,949     PRINT #13, "[Written on DATE: " + DATE$ + ", TIME: " + TIME$ + "].:"
00,950     PRINT #13,
00,951     'CLOSE #13
00,952     PRINT "The log/output file, opened in APPEND mode: "; CHR$(34); SHORTCUT$; CHR$(34)
00,953 $END IF
00,954
00,955 'f = FREEFILE
00,956 'SHELL _HIDE "cd > PRGMDIR.INF" 'get the current program path
00,957 'OPEN "PRGMDIR.INF" FOR INPUT AS #f
00,958 'LINE INPUT #f, current_program_path$
00,959 'CLOSE #f
00,960 'KILL "PRGMDIR.INF"
00,961 'PRINT "Current working directory path (obtained via SHELL _HIDE): "; CHR$(34); current_program_path$; CHR$(34)
00,962
00,963 'Ugh, they are reversed?!
00,964 'PRINT "Current working directory path: "; CHR$(34); _CWD$; CHR$(34)
00,965 'PRINT "User's program calling path: "; CHR$(34); _STARTDIR$; CHR$(34)
00,966 PRINT "Current working directory path: "; CHR$(34); _STARTDIR$; CHR$(34)
00,967 PRINT "User's program calling path: "; CHR$(34); _CWD$; CHR$(34)
00,968 'PRINT "Command line parameters sent when a program is started: "; CHR$(34); COMMAND$; CHR$(34)
00,969 'PRINT "_OS$="; _OS$
00,970
00,971 'Lazy me, in order to avoid editing the existing r.3+ code FileArray$() became function, array no more!
00,972 DIM FileArrayWINDOW$(YdimROW)
00,973
00,974 $IF WINDOWS THEN
00,975     _ACCEPTFILEDROP 'enables drag/drop functionality
00,976     IF COMMAND$ = "" THEN PRINT: PRINT "Drag files from a folder and drop them in this window... May press Alt+X or Alt+Q to eXit/Quit..."
00,977 $END IF
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,978
00,979 _DISPLAY
00,980
00,981 a$ = ""
00,982 PostfixedToHeader = CSRLIN
00,983 DO
00,984
00,985     LOCATE PostfixedToHeader, 1
00,986
00,987     'FOR d = 1 TO _DEVICES 'number of input devices found
00,988     '     dev$ = _DEVICE$(d)
00,989     '     IF INSTR(dev$, "[MOUSE]") THEN buttons = _LASTBUTTON(d); wheels = _LASTWHEEL(d); EXIT FOR
00,990     'NEXT
00,991     'PRINT
00,992     'PRINT "Mouse Buttons:"; buttons
00,993     'PRINT "Mouse Wheels:"; wheels
00,994     ''d = _DEVICES ' always read number of devices to enable device input
00,995     ''DO WHILE _DEVICEINPUT(2) 'loop only runs during a device 2 mouse event
00,996     ''LOOP
00,997
00,998     $IF WINDOWS THEN
00,999         ' MEMUSAGE by Steve [
01,000         '     GetCPULoad = 0 is idle, 1 is fully used.
01,001         '     Multiply by 100 for a percentage
01,002         PRINT
01,003         PRINT "CPU used:           "; LTRIM$(STR$(INT((GetCPULoad * 10000) / 100))); "%"; SPACE$(18)
01,004         PRINT "Memory used:           "; LTRIM$(STR$(MemInUsePercent)); "%"; SPACE$(18)
01,005         PRINT "Total Physical Memory: "; PRINT AddCommas$(TotalPhysicalMem); " bytes"; SPACE$(18)
01,006         PRINT "Free Physical Memory:  "; PRINT AddCommas$(FreePhysicalMem); " bytes"; SPACE$(18)
01,007         PRINT "Total Paging File:     "; PRINT AddCommas$(TotalPagingFile); " bytes"; SPACE$(18)
01,008         PRINT "Free Paging File:      "; PRINT AddCommas$(FreePagingFile); " bytes"; SPACE$(18)
01,009         '     PRINT "Total Virtual Memory: "; PRINT AddCommas$(TotalVirtualMem); " bytes"; SPACE$(18)
01,010         '     PRINT "Free Virtual Memory:  "; PRINT AddCommas$(FreeVirtualMem); " bytes"; SPACE$(18)
01,011         ' MEMUSAGE by Steve ]
01,012     $END IF
01,013
01,014     'IF COMMAND$ = "" THEN PRINT: PRINT "May press Alt+X or Alt+Q to eXit/Quit..."
01,015     ReturnCOMBO
01,016
01,017     IF YdimROW = 60 THEN
01,018         cryFrame = CSRLIN
01,019         NextFrame cryFrame + 1
01,020         LOCATE cryFrame, 1
01,021     END IF
01,022
01,023     _DISPLAY
01,024
01,025     $IF WINDOWS THEN
01,026         IF _TOTALDROPPEDFILES THEN

```

```

01,027 'FOR i = 1 TO _TOTALDROPPEDFILES
01,028 'a$ = _DROPPEDFILE(i)
01,029 a$ = _DROPPEDFILE(1)
01,030 'NEXT
01,031 _FINISHDROP 'If _FINISHDROP isn't called here then _TOTALDROPPEDFILES never gets reset.
01,032 'ELSE
01,033 'a$ = "Scroller.$$$"
01,034 'SHELL _HIDE "DIR /B *.* > Scroller.$$$"
01,035 END IF
01,036 $END IF
01,037 $IF WINDOWS THEN
01,038 IF COMMAND$ <> "" THEN a$ = _STARTDIR$ + "\" + COMMAND$
01,039 $ELSE
01,040 If Command$ <> "" Then a$ = _StartDir$ + "/" + Command$
01,041 $END IF
01,042
01,043 'WrapFlag = 0
01,044 'Wwidth% = XdimCOL
01,045 'ul& = 0
01,046 'LongestLine = 0
01,047 'DO WHILE NOT EOF(1)
01,048 ' LINE INPUT #1, l$: ul& = ul& + 1
01,049 ' ExpandTabs l$
01,050 ' IF LEN(l$) > LongestLine THEN LongestLine = LEN(l$)
01,051 ' IF WrapFlag THEN
01,052 ' lX$ = l$
01,053 ' DO WHILE LEN(lX$) > Wwidth%
01,054 ' Glupak% = Wwidth%
01,055 ' DO UNTIL MID$(lX$, Glupak% + 1, 1) = " " AND MID$(lX$, Glupak%, 1) <> " "
01,056 ' Glupak% = Glupak% - 1
01,057 ' IF Glupak% = 0 THEN
01,058 ' PRINT "Rejecting line(#"; LTRIM$(STR$(ul&)); ", "; LTRIM$(STR$(LEN(l$))); "chars) that cannot be wrapped!"
01,059 ' _DISPLAY: SYSTEM
01,060 ' GOTO B4Txpanar
01,061 ' END IF
01,062 ' LOOP
01,063 ' PRINT #3, LEFT$(lX$, Glupak%)
01,064 ' lX$ = STRING$(1, " ") + LTRIM$(MID$(lX$, Glupak% + 1, LEN(lX$) - (Glupak%)))
01,065 ' LOOP
01,066 ' PRINT #3, lX$
01,067 ' B4Txpanar:
01,068 ' END IF
01,069 'LOOP
01,070
01,071 SecondTime = 1
01,072 GettingStarted:
01,073
01,074 IF WrapFlag = 1 THEN
01,075 'Creating the wrapped temporary file [

```

```

01,076 IF _FILEEXISTS(a$ + ".wrapped") = 0 THEN
01,077 IF _FILEEXISTS(a$) THEN
01,078 IF SecondTime <> 2 THEN PRINT: PRINT "Writing wrapped variant..."
01,079 PostfixedToHeader = CSRLIN
01,080
01,081 StatuLine$ = SPACE$(XdimCOL)
01,082 MID$(StatuLine$, 1, 1) = "["
01,083 MID$(StatuLine$, XdimCOL, 1) = "]"
01,084
01,085 _DISPLAY
01,086 OPEN a$ FOR BINARY AS #1
01,087 OPEN a$ + ".wrapped" FOR OUTPUT AS #3
01,088 OPEN a$ + ".unwrappable" FOR OUTPUT AS #2
01,089 LongestLine = 0
01,090 UnwrappableLines = 0
01,091 FileSize = LOF(1)
01,092 IF FileSize THEN
01,093 ChunkBeingOneUnit = FileSize \ (XdimCOL - 1 - 1) + 1: LastPercentage = 0
01,094 NumberOfLFs = 0
01,095 Wwidth% = XdimCOL
01,096 ul% = 0
01,097 ReadBytes = 0
01,098 DO WHILE NOT EOF(1)
01,099 LINE INPUT #1, l$: ul% = ul% + 1: ReadBytes = ReadBytes + LEN(l$)
01,100 ExpandTabsFULL l$
01,101 IF LEN(l$) > LongestLine THEN LongestLine = LEN(l$)
01,102 IF WrapFlag THEN
01,103 ' Firstly do the wrapping virtually (in order to avoid writing some wrapped chunks and encounter unwrappable chunk) - we need either a wrapped line (in its entirety) or none:
01,104 AssumeLineIsWrappable = 1
01,105 lX$ = l$
01,106 DO WHILE LEN(lX$) > Wwidth%
01,107 Glupak% = Wwidth%
01,108 DO 'UNTIL (MID$(lX$, Glupak% + 1, 1) = " " OR MID$(lX$, Glupak% + 1, 1) = "\" OR MID$(lX$, Glupak% + 1, 1) = "/" OR MID$(lX$, Glupak% + 1, 1) = "_" OR MID$(lX$, Glupak% + 1, 1) = ";" OR
MID$(lX$, Glupak% + 1, 1) = ",") AND MID$(lX$, Glupak%, 1) <> " "
01,109 InvokeOnce$ = MID$(lX$, Glupak% + 1, 1)
01,110 IF INSTR(" \/_;.-!+%=:", InvokeOnce$) AND MID$(lX$, Glupak%, 1) <> " " THEN EXIT DO
01,111 'IF (InvokeOnce$ = " " OR InvokeOnce$ = "\" OR InvokeOnce$ = "/" OR InvokeOnce$ = "_" OR InvokeOnce$ = ";" OR InvokeOnce$ = "," OR InvokeOnce$ = "." OR InvokeOnce$ = "-" OR
InvokeOnce$ = "!") OR InvokeOnce$ = "+" OR InvokeOnce$ = "%") AND MID$(lX$, Glupak%, 1) <> " " THEN EXIT DO
01,112 Glupak% = Glupak% + 1
01,113 IF Glupak% = 0 THEN
01,114 AssumeLineIsWrappable = 0
01,115 GOTO B4TxpanarVIRTUAL
01,116 END IF
01,117 LOOP
01,118 lX$ = LTRIM$(MID$(lX$, Glupak% + 1, LEN(lX$) - (Glupak%)))
01,119 LOOP
01,120 B4TxpanarVIRTUAL:
01,121 IF AssumeLineIsWrappable = 1 THEN
01,122 lX$ = l$

```

```

01,123      DO WHILE LEN(1X$) > Wwidth%
01,124          Glupak% = Wwidth%
01,125          DO 'UNTIL (MID$(1X$, Glupak% + 1, 1) = " " OR MID$(1X$, Glupak% + 1, 1) = "\" OR MID$(1X$, Glupak% + 1, 1) = "/" OR MID$(1X$, Glupak% + 1, 1) = "_" OR MID$(1X$, Glupak% + 1, 1) = ";"
OR MID$(1X$, Glupak% + 1, 1) = ",") AND MID$(1X$, Glupak%, 1) <> " "
01,126          InvokeOnce$ = MID$(1X$, Glupak% + 1, 1)
01,127          IF INSTR(" \/_;.-!+%=: ", InvokeOnce$) AND MID$(1X$, Glupak%, 1) <> " " THEN EXIT DO
01,128          'IF (InvokeOnce$ = " " OR InvokeOnce$ = "\" OR InvokeOnce$ = "/" OR InvokeOnce$ = "_" OR InvokeOnce$ = ";" OR InvokeOnce$ = "," OR InvokeOnce$ = "." OR InvokeOnce$ = "-" OR
InvokeOnce$ = "! " OR InvokeOnce$ = "+" OR InvokeOnce$ = "%") AND MID$(1X$, Glupak%, 1) <> " " THEN EXIT DO
01,129          Glupak% = Glupak% - 1
01,130          IF Glupak% = 0 THEN
01,131              PRINT "Rejecting line(#"; LTRIM$(STR$(ul&)); ", "; LTRIM$(STR$(LEN(1$))); "chars) that cannot be wrapped!"
01,132              CLOSE #1, #3: KILL a$ + ".wrapped"
01,133              _DISPLAY: END
01,134              GOTO B4Txpanar
01,135          END IF
01,136          LOOP
01,137          PRINT #3, LEFT$(1X$, Glupak%)
01,138          '1X$ = STRING$(1, " ") + LTRIM$(MID$(1X$, Glupak% + 1, LEN(1X$) - (Glupak%)))
01,139          1X$ = LTRIM$(MID$(1X$, Glupak% + 1, LEN(1X$) - (Glupak%)))
01,140          LOOP
01,141          PRINT #3, 1X$
01,142          B4Txpanar:
01,143          ELSE
01,144              PRINT #2, 1$: UnwrappableLines = UnwrappableLines + 1
01,145          END IF
01,146          END IF
01,147          IF (ReadBytes \ ChunkBeingOneUnit) > LastPercentage THEN
01,148              LastPercentage = LastPercentage + 1 ' in range 1..(XdimCOL-1)
01,149              MID$(StatuLine$, 2, LastPercentage) = STRING$(LastPercentage, CHR$(176))
01,150              LOCATE YdimROW + 1, 1: COLOR 9, 0: PRINT StatuLine$;
01,151              _DISPLAY
01,152          END IF
01,153          LOOP
01,154          ELSE
01,155              CLOSE #1 'when filesize is 0
01,156              SYSTEM
01,157          END IF 'IF FileSize THEN
01,158          CLOSE #1, #3
01,159          a$ = a$ + ".wrapped"
01,160          LOCATE PostfixedToHeader, 1
01,161          COLOR NormalFRGr, NormalBCKGr
01,162          END IF
01,163          ELSE
01,164              a$ = a$ + ".wrapped"
01,165          END IF
01,166          'Creating the wrapped temporary file ]
01,167          END IF
01,168
01,169          TimeA = TIMER

```

```

01,170
01,171 ' Below fragment is r.4+ with 64bit pointers... to external/internal RAM!
01,172 ' On i5-7200u, SSD Kingston 256GB it loads OED in ???/??? seconds, respectively.
01,173 IF _FILEEXISTS(a$) THEN
01,174     IF SecondTime <> 2 THEN PRINT: PRINT "Loading...": PRINT
01,175
01,176     PostfixedToHeader = CSRLIN
01,177
01,178     StatuLine$ = SPACE$(XdimCOL)
01,179     MID$(StatuLine$, 1, 1) = "["
01,180     MID$(StatuLine$, XdimCOL, 1) = "]"
01,181
01,182     _DISPLAY
01,183     OPEN a$ FOR BINARY AS #1
01,184     LongestLine = 0
01,185     FileSize = LOF(1)
01,186     IF FileSize THEN
01,187         ChunkBeingOneUnit = FileSize \ (XdimCOL - 1 - 1) + 1: LastPercentage = 0
01,188         NumberOfLFs = 0
01,189         IF ToLoadOrNotFlag THEN
01,190             Mwholefile = _MEMNEW(FileSize + 1)
01,191             'TheWholeFile$ = SPACE$(FileSize + 1) ' Grmb1, this comes with limitation probably 2GB, which prompts for buffered e.g. 128KB approach as my old LineWordReporter.c tool.
01,192             'GET #1, , TheWholeFile$
01,193             ReadBytes = 0 ' Have to load the 2+GB "malloc" in chunks...
01,194             chunk128KB$ = SPACE$(128 * 1024 * 1024)
01,195             DO WHILE ReadBytes + (128 * 1024 * 1024) < FileSize
01,196                 GET #1, , chunk128KB$
01,197                 i = 1
01,198                 DO WHILE i <= (128 * 1024 * 1024)
01,199                     i = INSTR(i, chunk128KB$, CHR$(10))
01,200                     IF i = 0 THEN EXIT DO
01,201                     NumberOfLFs = NumberOfLFs + 1
01,202                     i = i + 1
01,203             LOOP
01,204             'FOR i = 1 TO LEN(chunk128KB$)
01,205             '     IF MID$(chunk128KB$, i, 1) = CHR$(10) THEN NumberOfLFs = NumberOfLFs + 1 ' Boost it with INSTR...
01,206             'NEXT i
01,207             _MEMPUT Mwholefile, Mwholefile.OFFSET + ReadBytes, chunk128KB$
01,208             ReadBytes = ReadBytes + (128 * 1024 * 1024)
01,209         LOOP
01,210         IF (FileSize - ReadBytes) THEN
01,211             RemainingChunk$ = SPACE$(FileSize - ReadBytes)
01,212             GET #1, , RemainingChunk$
01,213             i = 1
01,214             DO WHILE i <= LEN(RemainingChunk$)
01,215                 i = INSTR(i, RemainingChunk$, CHR$(10))
01,216                 IF i = 0 THEN EXIT DO
01,217                 NumberOfLFs = NumberOfLFs + 1
01,218                 i = i + 1

```

```

01,219         LOOP
01,220         'FOR i = 1 TO LEN(RemainingChunk$)
01,221         '   IF MID$(RemainingChunk$, i, 1) = CHR$(10) THEN NumberOfLFs = NumberOfLFs + 1 ' Boost it with INSTR...
01,222         'NEXT i
01,223         _MEMPUT Mwholefile, Mwholefile.OFFSET + ReadBytes, RemainingChunk$
01,224     END IF
01,225 END IF
01,226 'Bugfix from 2021-Jan-29, inhere problem exists, in efficient mode we don't know 'NumberOfLFs' therefore should read-it-without-loading-it:
01,227 IF ToLoadOrNotFlag = 0 THEN
01,228     ReadBytes = 0 ' Have to load the 2+GB "malloc" in chunks...
01,229     chunk128KB$ = SPACE$(128 * 1024 * 1024)
01,230     DO WHILE ReadBytes + (128 * 1024 * 1024) < FileSize
01,231         GET #1, , chunk128KB$
01,232         i = 1
01,233         DO WHILE i <= (128 * 1024 * 1024)
01,234             i = INSTR(i, chunk128KB$, CHR$(10))
01,235             IF i = 0 THEN EXIT DO
01,236             NumberOfLFs = NumberOfLFs + 1
01,237             i = i + 1
01,238         LOOP
01,239         ReadBytes = ReadBytes + (128 * 1024 * 1024)
01,240     LOOP
01,241     IF (FileSize - ReadBytes) THEN
01,242         RemainingChunk$ = SPACE$(FileSize - ReadBytes)
01,243         GET #1, , RemainingChunk$
01,244         i = 1
01,245         DO WHILE i <= LEN(RemainingChunk$)
01,246             i = INSTR(i, RemainingChunk$, CHR$(10))
01,247             IF i = 0 THEN EXIT DO
01,248             NumberOfLFs = NumberOfLFs + 1
01,249             i = i + 1
01,250         LOOP
01,251     END IF
01,252 END IF
01,253 SEEK #1, 1
01,254 'IF ASC(Byte) < 32 AND Byte <> CHR$(10) AND Byte <> CHR$(13) AND Byte <> CHR$(9) THEN Byte = CHR$(32)
01,255 filecount = 0
01,256 'Ugh, buggy, in r.5 below two "malloc" lines should be using 'NumberOfLFs' not 'FileSize'...
01,257 MhandleOFF = _MEMNEW(888 * (NumberOfLFs + 1)) 'create new memory block of 8*NumberOfLFs bytes - each line has its own 64bit Offset
01,258 MhandleLEN = _MEMNEW(888 * (NumberOfLFs + 1)) 'create new memory block of 8*NumberOfLFs bytes - each line has its own 64bit Length, kinda overkill, could be 32bit
01,259
01,260 ' Parser for r.6 [[[
01,261 '     LineLen13 = 0
01,262 '     Byte = ""
01,263 '     'Caution: Many files DON'T end with CRLF, or LF or CR - kinda the last line is not always postixed correctly!
01,264 '     'This means if after the last CRLF, or LF or CR there is at least one byte then the number of total lines should be +1
01,265 '     FOR j = 1 TO FileSize + 1
01,266 '     '
01,267 '         IF (j \ ChunkBeingOneUnit) > LastPercentage THEN

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

01,268      '      LastPercentage = LastPercentage + 1 ' in range 1..(XdimCOL-1)
01,269      '      MID$(StatuLine$, 2, LastPercentage) = STRING$(LastPercentage, CHR$(176))
01,270      '      LOCATE YdimROW + 1, 1: COLOR 9, 0: PRINT StatuLine$;
01,271      '      _DISPLAY
01,272      '      END IF
01,273      '
01,274      '      PrevByte = Byte
01,275      '      IF j <= FileSize THEN
01,276      '          IF ToLoadOrNotFlag THEN
01,277      '              'Byte = MID$(TheWholeFile$, j, 1)
01,278      '              _MEMGET Mwholefile, Mwholefile.OFFSET + (j - 1), Byte
01,279      '          ELSE
01,280      '              GET #1, , Byte
01,281      '          END IF
01,282      '      ELSE
01,283      '          Byte = ""
01,284      '          IF PrevByte <> CHR$(10) THEN Byte = CHR$(10) ' enforce last line to be postfixed ONLY if last char was not LF
01,285      '      END IF
01,286      '      QWORD = j
01,287      '      IF Byte = CHR$(10) THEN 'For now, won't work for MacOS...
01,288      '          QWORD = QWORD - LineLen13 'have to write the current offset minus the length of the line
01,289      '          _MEMPUT MhandleOFF, MhandleOFF.OFFSET + 888 * filecount, QWORD
01,290      '          IF PrevByte = CHR$(13) THEN LineLen13 = LineLen13 - 1
01,291      '          _MEMPUT MhandleLEN, MhandleLEN.OFFSET + 888 * filecount, LineLen13
01,292      '          IF LineLen13 > LongestLine THEN LongestLine = LineLen13
01,293      '          LineLen13 = 0
01,294      '          filecount = filecount + 1
01,295      '      ELSE
01,296      '          LineLen13 = LineLen13 + 1
01,297      '      END IF
01,298      '      NEXT j
01,299      ' Parser for r.6 ]]]
01,300
01,301      ' Parser for r.7 [[[[[[
01,302      IF ToLoadOrNotFlag = 0 THEN
01,303          ReadBytes = 0 ' Have to load the 2+GB "malloc" in chunks...
01,304          ChunkLen = 128 * 1024
01,305          chunk128KB$ = SPACE$(ChunkLen) ' +1 for sentinel
01,306          j = 1 ' is the current offset where new GET reads
01,307          QWORDlast = 1
01,308          PrevByte = ""
01,309          DO WHILE ReadBytes + (ChunkLen) < FileSize ' this '<' is important, on purpose not using '<=' since there should be a remnant chunk (where LF postfixing is enforced, eventually)
01,310
01,311              IF (j \ ChunkBeingOneUnit) > LastPercentage THEN
01,312
01,313                  IF SecondTime <> 2 THEN
01,314                      $IF WINDOWS THEN
01,315                          LOCATE PostfixedToHeader, 1
01,316                          ' MEMUSAGE by Steve [

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

01,317      ' GetCPULoad = 0 is idle, 1 is fully used.
01,318      ' Multiply by 100 for a percentage
01,319      PRINT "CPU used:      "; LTRIM$(STR$(INT((GetCPULoad * 10000) / 100))); "%"; SPACE$(18)
01,320      PRINT "Memory used:   "; LTRIM$(STR$(MemInUsePercent)); "%"; SPACE$(18)
01,321      PRINT "Total Physical Memory: "; PRINT AddCommas$(TotalPhysicalMem); " bytes"; SPACE$(18)
01,322      PRINT "Free Physical Memory: "; PRINT AddCommas$(FreePhysicalMem); " bytes"; SPACE$(18)
01,323      PRINT "Total Paging File: "; PRINT AddCommas$(TotalPagingFile); " bytes"; SPACE$(18)
01,324      PRINT "Free Paging File: "; PRINT AddCommas$(FreePagingFile); " bytes"; SPACE$(18)
01,325      PRINT "Total Virtual Memory: "; PRINT AddCommas$(TotalVirtualMem); " bytes"; SPACE$(18)
01,326      PRINT "Free Virtual Memory: "; PRINT AddCommas$(FreeVirtualMem); " bytes"; SPACE$(18)
01,327      ' MEMUSAGE by Steve ]
01,328      $END IF
01,329      END IF
01,330
01,331      LastPercentage = LastPercentage + 1 ' in range 1..(XdimCOL-1)
01,332      MID$(StatuLine$, 2, LastPercentage) = STRING$(LastPercentage, CHR$(176))
01,333      LOCATE YdimROW + 1, 1: COLOR 9, 0: PRINT StatuLine$;
01,334
01,335      IF YdimROW = 60 THEN
01,336          cryFrame = CSRLIN
01,337          NextFramePEN 1
01,338          LOCATE cryFrame, 1
01,339      END IF
01,340
01,341      _DISPLAY
01,342      END IF
01,343
01,344      LastByte = RIGHT$(chunk128KB$, 1) 'to handle eventual CR, left behind i.e. in previous chunk
01,345      GET #1, j, chunk128KB$
01,346      i = 1
01,347      FoundAt = INSTR(i, chunk128KB$, CHR$(10))
01,348      IF FoundAt THEN
01,349          DO WHILE FoundAt
01,350              IF FoundAt = 1 THEN PrevByte = LastByte ELSE PrevByte = MID$(chunk128KB$, FoundAt - 1, 1)
01,351              QWORD = (j - 1) + FoundAt
01,352              LineLen13 = QWORD - QWORDlast
01,353              _MEMPUT MhandleOFF, MhandleOFF.OFFSET + 888 * filecount, QWORDlast
01,354              QWORDlast = QWORD + 1
01,355              IF PrevByte = CHR$(13) THEN LineLen13 = LineLen13 - 1
01,356              _MEMPUT MhandleLEN, MhandleLEN.OFFSET + 888 * filecount, LineLen13
01,357              IF LineLen13 > LongestLine THEN LongestLine = LineLen13
01,358              filecount = filecount + 1
01,359              i = FoundAt + 1
01,360              IF i > (ChunkLen) THEN EXIT DO 'could use sentinel (buffer+1), in order this line to drop out
01,361              FoundAt = INSTR(i, chunk128KB$, CHR$(10))
01,362          LOOP
01,363      END IF
01,364      j = j + (ChunkLen)
01,365      ReadBytes = ReadBytes + (ChunkLen)

```

```

01,366      LOOP
01,367      IF (FileSize - ReadBytes) THEN
01,368          RemainingChunk$ = SPACE$(FileSize - ReadBytes) '*1 for sentinel
01,369          LastByte = RIGHT$(chunk128KB$, 1) 'to handle eventual CR, left behind i.e. in previous chunk
01,370          GET #1, , RemainingChunk$
01,371          IF RIGHT$(RemainingChunk$, 1) <> CHR$(10) THEN RemainingChunk$ = RemainingChunk$ + CHR$(10) ' dirty, enforcing not missing the last line (if it is not postfix with LF)
01,372          'Beware, yes be aware that above line should have been applied for above/first fragment because the filesize could be multiple of the chunk length i.e. no remaining chunk, however it was feinted by
01,373          '<'
01,374          i = 1
01,375          FoundAt = INSTR(i, RemainingChunk$, CHR$(10))
01,376          IF FoundAt THEN
01,377              DO WHILE FoundAt
01,378                  IF FoundAt = 1 THEN PrevByte = LastByte ELSE PrevByte = MID$(RemainingChunk$, FoundAt - 1, 1)
01,379                  QWORD = (j - 1) * FoundAt
01,380                  LineLen13 = QWORD - QWORDlast
01,381                  _MEMPUT MhandleOFF, MhandleOFF.OFFSET + 8&& * filecount, QWORDlast
01,382                  QWORDlast = QWORD + 1
01,383                  IF PrevByte = CHR$(13) THEN LineLen13 = LineLen13 - 1
01,384                  _MEMPUT MhandleLEN, MhandleLEN.OFFSET + 8&& * filecount, LineLen13
01,385                  IF LineLen13 > LongestLine THEN LongestLine = LineLen13
01,386                  filecount = filecount + 1
01,387                  i = FoundAt + 1
01,388                  IF i > (FileSize - ReadBytes) THEN EXIT DO 'could use sentinel (buffer+1), in order this line to drop out
01,389                  FoundAt = INSTR(i, RemainingChunk$, CHR$(10))
01,390              LOOP
01,391          END IF
01,392      END IF
01,393      SEEK #1, 1
01,394      ' Parser for r.7 ]]]]]]
01,395
01,396      ELSE
01,397          CLOSE #1 'when filesize is 0
01,398          SYSTEM
01,399      END IF 'IF FileSize THEN
01,400  END IF
01,401
01,402  ' Below fragment is r.4 with 64bit pointers... to external RAM!
01,403  ' On i5-7200u, SSD Kingston 256GB it loads OED in 3,339 seconds, naturally the second attempt is to read byte-by-byte from RAM, however the whole file has to be loaded.
01,404  'IF _FILEEXISTS(a$) THEN
01,405  '    PRINT: PRINT "Loading..."
01,406  '    _DISPLAY
01,407  '    OPEN a$ FOR BINARY AS #1
01,408  '    FileSize = LOF(1)
01,409  '    filecount = 0
01,410  '    MhandleOFF = _MEMNEW(8&& * FileSize) 'create new memory block of 8*filesize bytes - each line has its own 64bit Offset
01,411  '    MhandleLEN = _MEMNEW(8&& * FileSize) 'create new memory block of 8*filesize bytes - each line has its own 64bit Length, kinda overkill, could be 32bit
01,412  '    LineLen13 = 0
01,413  '    Byte = ""

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

01,414 ' 'Caution: Many files DON'T end with CRLF, or LF or CR - kinda the last line is not always postfixed correctly!
01,415 ' 'This means if after the last CRLF, or LF or CR there is at least one byte then the number of total lines should be +1
01,416 ' FOR j = 1 TO FileSize + 1
01,417 '     PrevByte = Byte
01,418 '     IF j <= FileSize THEN
01,419 '         GET #1, , Byte
01,420 '     ELSE
01,421 '         Byte = ""
01,422 '         IF PrevByte <> CHR$(10) THEN Byte = CHR$(10) ' enforce last line to be postfixed ONLY if last char was not LF
01,423 '     END IF
01,424 '     QWORD = j
01,425 '     IF Byte = CHR$(10) THEN 'For now, won't work for MacOS...
01,426 '         QWORD = QWORD - LineLen13 'have to write the current offset minus the length of the line
01,427 '         _MEMPUT MhandleOFF, MhandleOFF.OFFSET + 8&& * filecount, QWORD
01,428 '         IF PrevByte = CHR$(13) THEN LineLen13 = LineLen13 - 1
01,429 '         _MEMPUT MhandleLEN, MhandleLEN.OFFSET + 8&& * filecount, LineLen13
01,430 '         IF LineLen13 > LongestLine THEN LongestLine = LineLen13
01,431 '         LineLen13 = 0
01,432 '         filecount = filecount + 1
01,433 '     ELSE
01,434 '         LineLen13 = LineLen13 + 1
01,435 '     END IF
01,436 ' NEXT j
01,437 'END IF
01,438
01,439 ''PRINT "filecount= "; filecount
01,440 ''PRINT "LongestLine= "; LongestLine
01,441 ''exercise in reproducing the indexed file:
01,442 ''Debugo = CHR$(13) + CHR$(10)
01,443 'Debugo = CHR$(10)
01,444 'OPEN a$ + ".dump" FOR BINARY AS #2
01,445 'FOR i = 1 TO filecount
01,446 '     _MEMGET MhandleOFF, MhandleOFF.OFFSET + 8&& * (i - 1), QWORD
01,447 '     _MEMGET MhandleLEN, MhandleLEN.OFFSET + 8&& * (i - 1), LineLen13
01,448 '     PRINT QWORD, LineLen13:
01,449 '     SEEK #1, QWORD
01,450 '     BufferForLine$ = ""
01,451 '     FOR j = 1 TO LineLen13
01,452 '         GET #1, , Byte 'grmbl, how to load in one call?! As fread() is capable.
01,453 '         PUT #2, , Byte
01,454 '         BufferForLine$ = BufferForLine$ + Byte
01,455 '     NEXT j
01,456 '     PUT #2, , Debugo
01,457 '     PRINT BufferForLine$
01,458 'NEXT i
01,459 ' _DISPLAY
01,460 'END
01,461
01,462 ' Commented fragment is r.3+ with slow strings

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

01,463 'LongestLine = 0
01,464 'IF _FILEEXISTS(a$) THEN
01,465 '   PRINT: PRINT "Loading..."
01,466 '   _DISPLAY
01,467 '   filecount = 0
01,468 '   '       OPEN a$ FOR INPUT AS #1
01,469 '   OPEN a$ FOR BINARY AS #1 ' As suggested by SMcNeill and Pete in https://www.qb64.org/forum/index.php?topic=3518.msg128631#msg128631
01,470 '   DO UNTIL EOF(1)
01,471 '       LINE INPUT #1, l$
01,472 '       filecount = filecount + 1
01,473 '   LOOP
01,474 '   CLOSE #1
01,475 '   REDIM FileArray$(1 TO filecount)
01,476 '   filecount = 0
01,477 '   '       OPEN a$ FOR INPUT AS #1
01,478 '   OPEN a$ FOR BINARY AS #1 ' As suggested by SMcNeill and Pete in https://www.qb64.org/forum/index.php?topic=3518.msg128631#msg128631
01,479 '   FileSize = LOF(1)
01,480 '   DO UNTIL EOF(1)
01,481 '       LINE INPUT #1, l$
01,482 '       ExpandTabs l$
01,483 '       FOR j = 1 TO LEN(l$)
01,484 '           IF MID$(l$, j, 1) < CHR$(32) THEN MID$(l$, j, 1) = CHR$(32)
01,485 '       NEXT j
01,486 '       IF LEN(l$) > LongestLine THEN LongestLine = LEN(l$)
01,487 '       filecount = filecount + 1
01,488 '       FileArray$(filecount) = l$
01,489 '   LOOP
01,490 '   CLOSE #1
01,491 'END IF
01,492
01,493 _LIMIT 5
01,494 IF (_KEYDOWN(LALTkey&) OR _KEYDOWN(RALTkey&)) AND (_KEYDOWN(81) OR _KEYDOWN(113)) THEN SYSTEM
01,495 IF (_KEYDOWN(LALTkey&) OR _KEYDOWN(RALTkey&)) AND (_KEYDOWN(88) OR _KEYDOWN(120)) THEN SYSTEM
01,496
01,497 '           QB64 _KEYHIT and _KEYDOWN Values
01,498 '
01,499 'Esc F1  F2  F3  F4  F5  F6  F7  F8  F9  F10 F11 F12 Sys ScL Pause
01,500 ' 27 15104 15360 15616 15872 16128 16384 16640 16896 17152 17408 34048 34304 +316 +302 +019
01,501 '~ 1! 2@ 3# 4$ 5% 6^ 7& 8* 9( 0) _ += BkSp Ins Hme PUp NumL / * -
01,502 '126 33 64 35 36 37 94 38 42 40 41 95 43 8 20992 18176 18688 +300 47 42 45
01,503 ' 96 49 50 51 52 53 54 55 56 57 48 45 61
01,504 'Tab Q W E R T Y U I O P [{ } \| Del End Pdn 7Hme 8/? 9PU +
01,505 ' 9 81 87 69 82 84 89 85 73 79 80 123 125 124 21248 20224 20736 18176 18432 18688 43
01,506 ' 113 119 101 114 116 121 117 105 111 112 91 93 92 55 56 57
01,507 'CapL A S D F G H J K L ;: " Enter 4/?- 5 6/-?
01,508 '+301 65 83 68 70 71 72 74 75 76 58 34 13 19200 19456 19712 E
01,509 ' 97 115 100 102 103 104 106 107 108 59 39 52 53 54 n
01,510 'Shift Z X C V B N M ,< .> /? Shift ? 1End 2/? 3PD t
01,511 '+304 90 88 67 86 66 78 77 60 62 63 +303 18432 20224 20480 20736 e

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

01,512 '      122 120 99 118 98 110 109 44 46 47      49 50 51 r
01,513 'Ctrl Win Alt Spacebar Alt Win Menu Ctrl ?- ? -? 0Ins .Del
01,514 '+306 +311 +308 32 +307 +312 +319 +305 19200 20480 19712 20992 21248 13
01,515 '      48 46
01,516 '      Lower value = LCase/NumLock On _____ + = add 100000
01,517
01,518 LOOP WHILE a$ = ""
01,519
01,520 IF XdimCOL = 128 AND YdimROW = 40 THEN _TITLE "MASAKARI, revision 8.1+ 128x40, The 'Holy Axe' English Text Sidekick" + ": " + a$
01,521 IF XdimCOL = 128 AND YdimROW = 60 THEN _TITLE "MASAKARI, revision 8.1+ 128x60, The 'Holy Axe' English Text Sidekick" + ": " + a$
01,522 IF XdimCOL = 198 AND YdimROW = 60 THEN _TITLE "MASAKARI, revision 8.1+ 198x60, The 'Holy Axe' English Text Sidekick" + ": " + a$
01,523 IF XdimCOL = 300 AND YdimROW = 80 THEN _TITLE "MASAKARI, revision 8.1+ 300x80, The 'Holy Axe' English Text Sidekick" + ": " + a$
01,524 IF XdimCOL = 112 AND YdimROW = 30 THEN _TITLE "MASAKARI, revision 8.1+ 112x30, The 'Holy Axe' English Text Sidekick" + ": " + a$
01,525 LoadedFile$ = a$
01,526
01,527 TimeB = TIMER: SecondTime = 2
01,528 'PLAY "L8V2a-c-" 'removed in order to get rid of audio dependency
01,529 CurrentLine = 1
01,530 File_Frame_x = 1
01,531 File_Frame_y = 1
01,532 COLOR NormalFRGr, NormalBCKGr
01,533 UpdateWindowFrame NormalFRGr, NormalBCKGr
01,534
01,535 ' Here is the layout:
01,536 ' The window-frame is 1..crx or 1..128 | The file-frame is 1..LongestLine
01,537 '      .      .      |      .
01,538 '      .      .      |      .
01,539 '      cry      60      |      filecount
01,540 '      FileArrayWINDOW$(60) |      FileArray(filecount)
01,541 ' if File_Frame_x < 128 then PADDING to 128 else File_Frame_x = 1..LongestLine-(128-1)
01,542 ' if File_Frame_y < 60 then PADDING to 60 else File_Frame_y = 1..filecount-(60-1)
01,543
01,544 UpdateCLine 1, 1, InverseFRGr, InverseBCKGr, 1
01,545 StatuLine$ = SPACE$(XdimCOL)
01,546 MID$(StatuLine$, 1, 1) = "["
01,547 MID$(StatuLine$, XdimCOL, 1) = "]"
01,548
01,549 FieldLineNum = LEN(AddCommas$(filecount))
01,550 FieldLineLen = LEN(AddCommas$(LongestLine))
01,551
01,552 'Dumbo$ = "File Size: " + AddCommas$(FileSize) + "; Longest Line: " + AddCommas$(LongestLine)
01,553 Dumbo$ = "Filesize: " + AddCommas$(FileSize) + "; Longest: " + AddCommas$(LongestLine)
01,554 MID$(StatuLine$, 2, LEN(Dumbo$)) = Dumbo$
01,555 LOCATE YdimROW + 1, 1: COLOR 9, 0: PRINT StatuLine$;
01,556 UpdateCLL CurrentLine, LEN(Dumbo$) + 1 + 1 ' +1 due to [
01,557 'MostRightField = LEN(Dumbo$) + 1 + 1 + LEN("; Line Number: ") + LEN("; Line Length: ") + FieldLineNum + FieldLineLen
01,558 MostRightField = LEN(Dumbo$) + 1 + 1 + LEN("; Line: ") + LEN("; Linesize: ") + FieldLineNum + FieldLineLen
01,559 UpdateNextToCCL_DONE MostRightField
01,560

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

01,561 IF WrapFlag = 1 THEN
01,562     UpdateNextToCCL_UNWRAPPABLEcount MostRightField
01,563 END IF
01,564
01,565 _DISPLAY
01,566 LOCATE 1, 1, 1, 30, 31
01,567 crx = POS(0)
01,568 cry = CSRLIN
01,569 crxOLD = crx
01,570 cryOLD = cry
01,571 PageScrollBenchmark = 0
01,572 LineScrollBenchmark = 0
01,573
01,574 tCurrentT1# = TIMER(.001)
01,575 tLASTrelease1# = tCurrentT1#
01,576 tLASTpress1# = tCurrentT1#
01,577 tCurrentT2# = TIMER(.001)
01,578 tLASTrelease2# = tCurrentT2#
01,579 tLASTpress2# = tCurrentT2#
01,580
01,581 DO 'mainloop
01,582     ReturnCOMBO
01,583     IsF12released_Flag = 0
01,584     IsF2released_Flag = 0
01,585     IsF3released_Flag = 0
01,586     IsENTERreleased_Flag = 0
01,587
01,588     MustBeLONGssigned& = _KEYHIT
01,589     IF MustBeLONGssigned& THEN
01,590         IF MustBeLONGssigned& < 0 THEN 'negative value means key released
01,591             MustBeLONGssigned& = -MustBeLONGssigned&
01,592             IF MustBeLONGssigned& \ 256 = 60 THEN
01,593                 IsF2released_Flag = 1
01,594             ELSE
01,595                 END IF
01,596             IF MustBeLONGssigned& \ 256 = 61 THEN
01,597                 IsF3released_Flag = 1
01,598             ELSE
01,599                 END IF
01,600             IF MustBeLONGssigned& \ 256 = 134 THEN
01,601                 IsF12released_Flag = 1
01,602             ELSE
01,603                 END IF
01,604             IF MustBeLONGssigned& = 13 THEN
01,605                 IsENTERreleased_Flag = 1
01,606             ELSE
01,607                 END IF
01,608             END IF
01,609     END IF

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

01,659      OPEN "masakari.wrd" FOR BINARY AS #6
01,660      IF LOF(6) THEN
01,661          CLOSE #6
01,662          OPEN "masakari.ind" FOR RANDOM ACCESS READ AS #6 LEN = 32
01,663          LeftPoint& = 1
01,664          RightPoint& = TotalWrd&
01,665          DO
01,666              IF RightPoint& - LeftPoint& <= 1 THEN 'Sxsedtcheta
01,667                  GET #6, LeftPoint&, lfixed$
01,668                  IF wrd$ = UCASE$(RTRIM$(lfixed$)) THEN WRDpresent% = 1
01,669                  GET #6, RightPoint&, lfixed$
01,670                  IF wrd$ = UCASE$(RTRIM$(lfixed$)) THEN WRDpresent% = 1
01,671                  EXIT DO
01,672              END IF
01,673              SplitPoint& = (LeftPoint& + RightPoint&) \ 2
01,674              GET #6, SplitPoint&, lfixed$
01,675              IF wrd$ < UCASE$(RTRIM$(lfixed$)) THEN
01,676                  RightPoint& = SplitPoint&
01,677              ELSEIF wrd$ > UCASE$(RTRIM$(lfixed$)) THEN
01,678                  LeftPoint& = SplitPoint&
01,679              ELSE
01,680                  WRDpresent% = 1: EXIT DO
01,681              END IF
01,682          LOOP
01,683          CLOSE #6
01,684      ELSE
01,685          CLOSE #6
01,686          KILL "masakari.wrd"
01,687          WRDpresent% = 2
01,688      END IF
01,689      IF WRDpresent% = 0 THEN
01,690          FOR Underdog% = j% - LEN(wrd$) + 1 TO j%
01,691              MID$(BigMoron$, Underdog%, 1) = "#"
01,692          NEXT
01,693      END IF
01,694
01,695      wrd$ = ""
01,696      Nothing2:
01,697      LOOP WHILE j% < Matroz%
01,698
01,699      FOR j% = 1 TO XdimCOL
01,700          IF MID$(BigMoron$, j%, 1) <> " " THEN
01,701              IF i = InverseL THEN ' inverse line
01,702                  LOCATE i, j%: COLOR 15, InverseBCKGr: PRINT MID$(Moron$, j%, 1);
01,703              ELSE
01,704                  LOCATE i, j%: COLOR 15, 0: PRINT MID$(Moron$, j%, 1);
01,705              END IF
01,706          END IF
01,707      NEXT

```

Listing: MASAKARI Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

01,757      KeyTap_Button1LOG_firstDetection#(4) = KeyTap_PrevClick1#
01,758      KeyTap_Button1LOG_ForHowLongHoled#(4) = KeyTap_ForHowLongWasPressed1#
01,759      KeyTap_PrevClick1# = 0
01,760  END IF
01,761
01,762  IF KeyTap_buttondown2 = 0 AND KeyTap_PrevClick2# <> 0 THEN 'write to the log
01,763      FOR MumboJumbo = 1 TO 3
01,764          KeyTap_Button2LOG_firstDetection#(MumboJumbo) = KeyTap_Button2LOG_firstDetection#(MumboJumbo + 1)
01,765          KeyTap_Button2LOG_ForHowLongHoled#(MumboJumbo) = KeyTap_Button2LOG_ForHowLongHoled#(MumboJumbo + 1)
01,766      NEXT
01,767      KeyTap_Button2LOG_firstDetection#(4) = KeyTap_PrevClick2#
01,768      KeyTap_Button2LOG_ForHowLongHoled#(4) = KeyTap_ForHowLongWasPressed2#
01,769      KeyTap_PrevClick2# = 0
01,770  END IF
01,771
01,772  IF KeyTap_buttondown3 = 0 AND KeyTap_PrevClick3# <> 0 THEN 'write to the log
01,773      FOR MumboJumbo = 1 TO 3
01,774          KeyTap_Button3LOG_firstDetection#(MumboJumbo) = KeyTap_Button3LOG_firstDetection#(MumboJumbo + 1)
01,775          KeyTap_Button3LOG_ForHowLongHoled#(MumboJumbo) = KeyTap_Button3LOG_ForHowLongHoled#(MumboJumbo + 1)
01,776      NEXT
01,777      KeyTap_Button3LOG_firstDetection#(4) = KeyTap_PrevClick3#
01,778      KeyTap_Button3LOG_ForHowLongHoled#(4) = KeyTap_ForHowLongWasPressed3#
01,779      KeyTap_PrevClick3# = 0
01,780  END IF
01,781
01,782  IF KeyTap_buttondown4 = 0 AND KeyTap_PrevClick4# <> 0 THEN 'write to the log
01,783      FOR MumboJumbo = 1 TO 3
01,784          KeyTap_Button4LOG_firstDetection#(MumboJumbo) = KeyTap_Button4LOG_firstDetection#(MumboJumbo + 1)
01,785          KeyTap_Button4LOG_ForHowLongHoled#(MumboJumbo) = KeyTap_Button4LOG_ForHowLongHoled#(MumboJumbo + 1)
01,786      NEXT
01,787      KeyTap_Button4LOG_firstDetection#(4) = KeyTap_PrevClick4#
01,788      KeyTap_Button4LOG_ForHowLongHoled#(4) = KeyTap_ForHowLongWasPressed4#
01,789      KeyTap_PrevClick4# = 0
01,790  END IF
01,791
01,792  IF KeyTap_Button1LOG_firstDetection#(3) AND (KeyTap_Button1LOG_firstDetection#(4) - KeyTap_Button1LOG_firstDetection#(3) < DclickTime#) THEN Double_LSHIFTkey = 1 ELSE Double_LSHIFTkey = 0
01,793  IF KeyTap_Button2LOG_firstDetection#(3) AND (KeyTap_Button2LOG_firstDetection#(4) - KeyTap_Button2LOG_firstDetection#(3) < DclickTime#) THEN Double_LCTRLkey = 1 ELSE Double_LCTRLkey = 0
01,794  IF KeyTap_Button3LOG_firstDetection#(3) AND (KeyTap_Button3LOG_firstDetection#(4) - KeyTap_Button3LOG_firstDetection#(3) < DclickTime#) THEN Double_LALTkey = 1 ELSE Double_LALTkey = 0
01,795  IF KeyTap_Button4LOG_firstDetection#(3) AND (KeyTap_Button4LOG_firstDetection#(4) - KeyTap_Button4LOG_firstDetection#(3) < DclickTime#) THEN Double_RALTkey = 1 ELSE Double_RALTkey = 0
01,796  ' keyboard mumbo-jumbo - handling of double-hits ]
01,797
01,798  IF Double_LSHIFTkey = 1 THEN
01,799      Double_LSHIFTkey = 0 'clear from buffer
01,800      FOR MumboJumbo = 1 TO 4
01,801          KeyTap_Button1LOG_firstDetection#(MumboJumbo) = 0
01,802          KeyTap_Button1LOG_ForHowLongHoled#(MumboJumbo) = 0
01,803      NEXT
01,804      crxInfo = POS(0)
01,805      cryInfo = CSRLIN

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

01,806 LOCATE YdimROW + 1, MostRightField: COLOR 4, 0: PRINT SPACE$((XdimCOL - MostRightField) - 0);
01,807 LOCATE YdimROW + 1, MostRightField: COLOR 4, 0: PRINT "; Pressed 'Double_LSHIFTkey', but unassigned.";
01,808 LOCATE cryInfo, crxInfo, 1, 30, 31
01,809 END IF
01,810 IF Double_LCTRLkey = 1 THEN
01,811 Double_LCTRLkey = 0 'clear from buffer
01,812 FOR MumboJumbo = 1 TO 4
01,813 KeyTap_Button2LOG_firstDetection#(MumboJumbo) = 0
01,814 KeyTap_Button2LOG_ForHowLongHoled#(MumboJumbo) = 0
01,815 NEXT
01,816 crxInfo = POS(0)
01,817 cryInfo = CSRLIN
01,818 LOCATE YdimROW + 1, MostRightField: COLOR 4, 0: PRINT SPACE$((XdimCOL - MostRightField) - 0);
01,819 LOCATE YdimROW + 1, MostRightField: COLOR 4, 0: PRINT "; Pressed 'Double_LCTRLkey', but unassigned.";
01,820 LOCATE cryInfo, crxInfo, 1, 30, 31
01,821 END IF
01,822 IF Double_LALTkey = 1 THEN
01,823 Double_LALTkey = 0 'clear from buffer
01,824 FOR MumboJumbo = 1 TO 4
01,825 KeyTap_Button3LOG_firstDetection#(MumboJumbo) = 0
01,826 KeyTap_Button3LOG_ForHowLongHoled#(MumboJumbo) = 0
01,827 NEXT
01,828 crxInfo = POS(0)
01,829 cryInfo = CSRLIN
01,830 LOCATE YdimROW + 1, MostRightField: COLOR 4, 0: PRINT SPACE$((XdimCOL - MostRightField) - 0);
01,831 LOCATE YdimROW + 1, MostRightField: COLOR 4, 0: PRINT "; Pressed 'Double_LALTkey', but unassigned.";
01,832 LOCATE cryInfo, crxInfo, 1, 30, 31
01,833 END IF
01,834 IF Double_RALTkey = 1 THEN
01,835 Double_RALTkey = 0 'clear from buffer
01,836 FOR MumboJumbo = 1 TO 4
01,837 KeyTap_Button4LOG_firstDetection#(MumboJumbo) = 0
01,838 KeyTap_Button4LOG_ForHowLongHoled#(MumboJumbo) = 0
01,839 NEXT
01,840 crxInfo = POS(0)
01,841 cryInfo = CSRLIN
01,842 LOCATE YdimROW + 1, MostRightField: COLOR 4, 0: PRINT SPACE$((XdimCOL - MostRightField) - 0);
01,843 LOCATE YdimROW + 1, MostRightField: COLOR 4, 0: PRINT "; Pressed 'Double_RALTkey', but unassigned.";
01,844 LOCATE cryInfo, crxInfo, 1, 30, 31
01,845 END IF
01,846
01,847 IF _KEYDOWN(LSHIFTkey&) = 0 AND _KEYDOWN(15616) AND _KEYDOWN(LCTRLkey&) THEN 'Lctrl+F3
01,848 'CALL DrawBox(3, 3, 7, 44, 8, 7)
01,849 'CALL DrawBoxShadow(3, 3, 7, 44)
01,850 'In case of YdimROW = 40 then shrunk panel
01,851 IF YdimROW = 40 THEN shrunkP = 14 ELSE shrunkP = 0
01,852 CALL DrawBoxShadow3(3, 3, 48 - shrunkP, 107, "Lctrl+F3: Search Panel")
01,853 UpdateWindowFrame NormalFRGr, NormalBCKGr
01,854 _DISPLAY

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

01,855      IsF3released_Flag = 0
01,856      KEYCLEAR
01,857  END IF
01,858
01,859  'IF _KEYDOWN(LSHIFTkey&) = 0 AND _KEYDOWN(15616) AND _KEYDOWN(LALTkey&) THEN 'LAlt+F3released
01,860  IF _KEYDOWN(LSHIFTkey&) = 0 AND _KEYDOWN(15616) AND _KEYDOWN(LALTkey&) = 0 THEN 'F3released
01,861      DO WHILE _KEYDOWN(15616): LOCATE , , 0: _DISPLAY: LOOP
01,862      'CALL DrawBox(3, 3, 7, 44, 8, 7)
01,863      'CALL DrawBoxShadow(3, 3, 7, 44)
01,864      'In case of YdimROW = 40 then shrunk panel
01,865      shrunkP = 14
01,866      'SUB DrawBoxShadow3 (TopLrow, TopLcol, BottomRow, BottomRcol, Captnn$)
01,867      F3pattern$ = DrawBoxShadow3exact$(3 + 13, 3, 35 - shrunkP, 107, "F3: Exact Case-Sensitive Search Panel")
01,868      DO UNTIL IsENTERreleased: LOCATE , , 0: _DISPLAY: LOOP 'doesnt work on Linux Fedora 33
01,869      UpdateWindowFrame NormalFRGr, NormalBCKGr
01,870      UpdateNextToOCL_F3 MostRightField
01,871      _DISPLAY
01,872
01,873  DO WHILE CurrentLine < filecount
01,874      ' Simulating Dn [
01,875      'CASE 80:
01,876      IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
01,877      IF cry < YdimROW THEN
01,878          IF cry < filecount THEN cry = cry + 1 'down
01,879      ELSE 'scrolling is needed
01,880          IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
01,881          'UpdateWindowFrame NormalFRGr, NormalBCKGr
01,882      END IF
01,883      ' Simulating Dn ]
01,884      TheNewCL$ = FileArray$(CurrentLine)
01,885      IF INSTR(TheNewCL$, F3pattern$) THEN
01,886          UpdateWindowFrame NormalFRGr, NormalBCKGr
01,887          _DISPLAY
01,888          EXIT DO
01,889      ELSE
01,890          IF CurrentLine = filecount THEN
01,891              UpdateWindowFrame NormalFRGr, NormalBCKGr
01,892              _DISPLAY
01,893              $IF WINDOWS THEN
01,894                  PLAY "v20132c"
01,895              $END IF
01,896              EXIT DO
01,897          END IF
01,898      END IF
01,899  LOOP
01,900
01,901  ' Interactive search, from the line below the INVERSE one, forward ... ]
01,902  UpdateNextToOCL_DONE MostRightField
01,903  IsF3released_Flag = 0

```

```

01,904      _KEYCLEAR
01,905  END IF
01,906
01,907  'IF _KEYDOWN(LSHIFTkey&) = 0 AND _KEYDOWN(34304) AND _KEYDOWN(LALTkey&) THEN 'LAlt+F12released
01,908  IF _KEYDOWN(LSHIFTkey&) = 0 AND _KEYDOWN(15360) AND _KEYDOWN(LALTkey&) = 0 THEN 'F2released
01,909    DO WHILE _KEYDOWN(15360): LOCATE , , 0: _DISPLAY: LOOP
01,910    'CALL DrawBox(3, 3, 7, 44, 8, 7)
01,911    'CALL DrawBoxShadow(3, 3, 7, 44)
01,912    'In case of YdimROW = 40 then shrunk panel
01,913    shrunkP = 14
01,914    'SUB DrawBoxShadow3 (TopLrow, TopLcol, BottomRow, BottomRcol, Captnn$)
01,915    F3pattern$ = DrawBoxShadow3exactRussian$(3 + 13, 3, 35 - shrunkP, 107, "F2: Exact Case-Sensitive Search Panel (Russian Phonetic)")
01,916    DO UNTIL IsENTERreleased: LOCATE , , 0: _DISPLAY: LOOP 'doesnt work on Linux Fedora 33
01,917    UpdateWindowFrame NormalFRGr, NormalBCKGr
01,918    UpdateNextToOCL_F3 MostRightField
01,919    _DISPLAY
01,920
01,921  DO WHILE CurrentLine < filecount
01,922    ' Simulating Dn [
01,923    'CASE 80:
01,924    IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
01,925    IF cry < YdimROW THEN
01,926      IF cry < filecount THEN cry = cry + 1 'down
01,927    ELSE 'scrolling is needed
01,928      IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
01,929      'UpdateWindowFrame NormalFRGr, NormalBCKGr
01,930    END IF
01,931    ' Simulating Dn ]
01,932    TheNewCL$ = FileArray$(CurrentLine)
01,933    IF INSTR(TheNewCL$, F3pattern$) THEN
01,934      UpdateWindowFrame NormalFRGr, NormalBCKGr
01,935      _DISPLAY
01,936      EXIT DO
01,937    ELSE
01,938      IF CurrentLine = filecount THEN
01,939        UpdateWindowFrame NormalFRGr, NormalBCKGr
01,940        _DISPLAY
01,941        $IF WINDOWS THEN
01,942          PLAY "v20132c"
01,943        $END IF
01,944        EXIT DO
01,945      END IF
01,946    END IF
01,947  LOOP
01,948
01,949  ' Interactive search, from the line below the INVERSE one, forward ... ]
01,950  UpdateNextToOCL_DONE MostRightField
01,951  IsF2released_Flag = 0
01,952  _KEYCLEAR

```

```

01,953     END IF
01,954
01,955 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) = 0 AND _KEYDOWN(15616) THEN 'LShift+(F3 released)
01,956 DO WHILE _KEYDOWN(15616): LOCATE , , 0: _DISPLAY: LOOP
01,957 'DO UNTIL IsF3released: LOCATE , , 0: _DISPLAY: LOOP ' doesn't work on Linux Fedora 33
01,958 UpdateNextToOCL_F3 MostRightField
01,959 _DISPLAY
01,960 IF INSTR(_CLIPBOARD$, CHR$(10)) THEN '_CLIPBOARD$ could be multi-line, so get the first
01,961 a2$ = LEFT$(_CLIPBOARD$, INSTR(_CLIPBOARD$, CHR$(10)) - 1)
01,962 IF RIGHT$(a2$, 1) = CHR$(13) THEN a2$ = LEFT$(a2$, LEN(a2$) - 1)
01,963 F3pattern$ = a2$
01,964 ELSE
01,965 F3pattern$ = _CLIPBOARD$
01,966 END IF
01,967 ' Interactive search, from the line below the INVERSE one, forward ... [
01,968 DO WHILE CurrentLine < filecount
01,969 ' Simulating Dn [
01,970 'CASE 80:
01,971 IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
01,972 IF cry < YdimROW THEN
01,973 IF cry < filecount THEN cry = cry + 1 'down
01,974 ELSE 'scrolling is needed
01,975 IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
01,976 'UpdateWindowFrame NormalFRGr, NormalBCKGr
01,977 END IF
01,978 ' Simulating Dn ]
01,979 TheNewCL$ = FileArray$(CurrentLine)
01,980 IF INSTR(TheNewCL$, F3pattern$) THEN
01,981 UpdateWindowFrame NormalFRGr, NormalBCKGr
01,982 _DISPLAY
01,983 EXIT DO
01,984 ELSE
01,985 IF CurrentLine = filecount THEN
01,986 UpdateWindowFrame NormalFRGr, NormalBCKGr
01,987 _DISPLAY
01,988 $IF WINDOWS THEN
01,989 PLAY "v20132c"
01,990 $END IF
01,991 EXIT DO
01,992 END IF
01,993 END IF
01,994 LOOP
01,995
01,996 ' Interactive search, from the line below the INVERSE one, forward ... ]
01,997 UpdateNextToOCL_DONE MostRightField
01,998 END IF
01,999
02,000 IF _KEYDOWN(LSHIFTkey&) = 0 AND _KEYDOWN(LCTRLkey&) = 0 AND _KEYDOWN(15616) AND _KEYDOWN(LALTkey&) THEN 'LAlt+F3released
02,001 DO WHILE _KEYDOWN(15616): LOCATE , , 0: _DISPLAY: LOOP

```

```

02,002      'DO UNTIL IsF3released: LOCATE , , 0: _DISPLAY: LOOP ' doesn't work on Linux Fedora 33
02,003      UpdateNextToOCL_F3 MostRightField
02,004      _DISPLAY
02,005      F3pattern$ = CurrentWord$
02,006
02,007      ' Interactive search, from the line below the INVERSE one, forward ... [
02,008
02,009      DO WHILE CurrentLine < filecount
02,010          ' Simulating Dn [
02,011          'CASE 80:
02,012          IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,013          IF cry < YdimROW THEN
02,014              IF cry < filecount THEN cry = cry + 1 'down
02,015          ELSE 'scrolling is needed
02,016              IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
02,017              'UpdateWindowFrame NormalFBGr, NormalBCKGr
02,018          END IF
02,019          ' Simulating Dn ]
02,020          TheNewCL$ = FileArray$(CurrentLine)
02,021          IF INSTR(TheNewCL$, F3pattern$) THEN
02,022              UpdateWindowFrame NormalFBGr, NormalBCKGr
02,023              _DISPLAY
02,024              EXIT DO
02,025          ELSE
02,026              IF CurrentLine = filecount THEN
02,027                  UpdateWindowFrame NormalFBGr, NormalBCKGr
02,028                  _DISPLAY
02,029                  $IF WINDOWS THEN
02,030                      PLAY "v20132c"
02,031                  $END IF
02,032                  EXIT DO
02,033              END IF
02,034          END IF
02,035      LOOP
02,036
02,037      ' Interactive search, from the line below the INVERSE one, forward ... ]
02,038      UpdateNextToOCL_DONE MostRightField
02,039      END IF
02,040
02,041      IF _KEYDOWN(27) THEN
02,042          LineScrollBenchmark = 0
02,043          PageScrollBenchmark = 0
02,044      END IF
02,045
02,046      IF LSHIFT_RSHIFT OR LineScrollBenchmark = 1 THEN 'same as scroll page-by-page benchmark, but line-by-line
02,047          IF LineScrollBenchmark = 0 THEN TimeScrollA = TIMER
02,048          LineScrollBenchmark = 1
02,049          'PgDn - just add the page height i.e. 'YdimROW' to 'File_Frame_y'
02,050          ' Don't execute PgDn (advancing the 'CurrentLine') if 'File_Frame_y' is not "eligible":

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

02,051      'IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
02,052      'IF File_Frame_y + 1 <= filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
02,053      'CAUTION [
02,054      ' Next three line beep NOT
02,055      'DEFLNG A-Z
02,056      'aaa = 61
02,057      'bbb = -48
02,058      'IF aaa <= bbb THEN BEEP: END
02,059      'Next three line beep
02,060      'aaa~& = 61
02,061      'bbb~& = -48
02,062      'IF aaa~& <= bbb~& THEN BEEP: END
02,063      'Next three line beep NOT
02,064      'aaa& = 61
02,065      'bbb& = -48
02,066      'IF aaa& <= bbb& THEN BEEP: END
02,067      'Next line works even when unsigned!
02,068      'IF File_Frame_y + YdimROW - (filecount - (YdimROW - 1)) <= 0 THEN
02,069      'CAUTION ]
02,070      'Next line doesn't work when unsigned!
02,071      IF File_Frame_y + 1 <= filecount - (YdimROW - 1) THEN
02,072          File_Frame_y = File_Frame_y + 1
02,073          'IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,074          'IF CurrentLine + 1 <= filecount THEN CurrentLine = CurrentLine + 1
02,075          IF CurrentLine + 1 <= filecount THEN CurrentLine = CurrentLine + 1
02,076          UpdateWindowFrame NormalFRGr, NormalBCKGr
02,077      ELSE
02,078          IF LineScrollBenchmark = 1 THEN
02,079              TimeScrollB = TIMER
02,080              _KEYCLEAR
02,081              ReportTimeToScroll MostRightField
02,082              LineScrollBenchmark = 0
02,083              _DISPLAY
02,084          END IF
02,085      END IF
02,086  END IF
02,087
02,088  ' Esc F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 Sys ScL Pause
02,089  ' 27 +59 +60 +61 +62 +63 +64 +65 +66 +67 +68 +133 +134 - - -
02,090  ' `` 1! 2@ 3# 4$ 5% 6^ 7& 8* 9( 0) - =+ BkSp Ins Hme PUp NumL / * -
02,091  ' 126 33 64 35 36 37 94 38 42 40 41 95 43 8 +82 +71 +73 - 47 42 45
02,092  ' 96 49 50 51 52 53 54 55 56 57 48 45 61
02,093  ' Tab Q W E R T Y U I O P [{ ]} \| Del End PDn 7Hme 8/ 9PU +
02,094  ' 9 81 87 69 82 84 89 85 73 79 80 123 125 124 +83 +79 +81 +71 +72 +73 43
02,095  ' 113 119 101 114 116 121 117 105 111 112 91 93 92 55 56 57
02,096  ' CapL A S D F G H J K L ;: '" Enter 4/- 5 6/-
02,097  ' - 65 83 68 70 71 72 74 75 76 58 34 13 +75 +76 +77 E
02,098  ' 97 115 100 102 103 104 106 107 108 59 39 52 53 54 n
02,099  '

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

02,100 ' Shift Z X C V B N M , < . > / ? Shift 1End 2/ 3PD t
02,101 ' * 90 88 67 86 66 78 77 60 62 63 * +72 +79 +80 +81 e
02,102 ' 122 120 99 118 98 110 109 44 46 47 49 50 51 r
02,103 ' Ctrl Win Alt Spacebar Alt Win Menu Ctrl - - 0Ins .Del
02,104 ' * - * 32 * - - * +75 +80 +77 +82 +83 13
02,105 ' 48 46
02,106 '
02,107 ' Italics = LCase/NumLock On _____ + = 2 Byte: CHR$(0) + CHR$(code)
02,108 'NOTE: The above commented table can be copied and pasted directly into the QB64 IDE
02,109
02,110 key$ = INKEY$
02,111 'DO: a$ = INKEY$; LOOP UNTIL a$ <> "" ' prevent ASC empty string read error
02,112 IF key$ <> "" THEN
02,113 code% = ASC(key$);
02,114 IF code% THEN ' ASC returns any value greater than 0
02,115 SELECT CASE ASC(key$)
02,116 CASE 32:
02,117 a$ = RTRIM$(FileArray$(CurrentLine))
02,118 IF _FILEEXISTS(a$) THEN
02,119 CLOSE #1
02,120 _MEMFREE MhandleOFF
02,121 _MEMFREE MhandleLEN
02,122 IF ToLoadOrNotFlag THEN
02,123 _MEMFREE Mwholefile
02,124 END IF
02,125 GOTO GettingStarted 'Before going above lines must be executed, i.e. to initialize.
02,126 END IF
02,127 CASE 65 TO 97: 'PRINT key$;
02,128 CASE ASC("a") TO ASC("z"): 'PRINT key$;
02,129 'CASE 27: COLOR 7, 0: SYSTEM 'END
02,130 END SELECT
02,131 ELSE
02,132 SELECT CASE ASC(key$, 2)
02,133 CASE 59:
02,134 LOCATE 1, 1, 0, 30, 31
02,135 FOR i = 1 TO YdimROW
02,136 LOCATE i, 1: PRINT SPACE$(XdimCOL);
02,137 NEXT
02,138 LOCATE 1, 1
02,139 ShowF1
02,140 PRINT "Press ESC...";
02,141 _DISPLAY
02,142 $IF WINDOWS THEN
02,143 PLAY "v20120g"
02,144 $END IF
02,145 DO WHILE INKEY$ <> CHR$(27)
02,146 _LIMIT 30
02,147 LOOP
02,148 UpdateWindowFrame NormalFRGr, NormalBCKGr

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

02,149     CASE 72:
02,150         IF CurrentLine > 1 THEN CurrentLine = CurrentLine - 1
02,151         IF cry > 1 THEN
02,152             cry = cry - 1 'up
02,153         ELSE 'scrolling is needed
02,154             IF File_Frame_y > 1 THEN File_Frame_y = File_Frame_y - 1
02,155             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,156         END IF
02,157     CASE 80:
02,158         IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,159         IF cry < YdimROW THEN
02,160             IF cry < filecount THEN cry = cry + 1 'down
02,161         ELSE 'scrolling is needed
02,162             IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
02,163             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,164         END IF
02,165     CASE 75: IF crx > 1 THEN crx = crx - 1 'left
02,166     CASE 77: IF crx < XdimCOL THEN crx = crx + 1 'right
02,167     CASE 73: 'PgUp
02,168         IF File_Frame_y - YdimROW >= 1 THEN
02,169             File_Frame_y = File_Frame_y - YdimROW
02,170             IF CurrentLine - YdimROW >= 1 THEN CurrentLine = CurrentLine - YdimROW
02,171             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,172         END IF
02,173     CASE 81: 'PgDn
02,174         IF File_Frame_y + YdimROW <= filecount - (YdimROW - 1) THEN
02,175             File_Frame_y = File_Frame_y + YdimROW
02,176             'IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,177             'IF CurrentLine + 1 <= filecount THEN CurrentLine = CurrentLine + 1
02,178             IF CurrentLine + YdimROW <= filecount THEN CurrentLine = CurrentLine + YdimROW
02,179             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,180         END IF
02,181     END SELECT
02,182     END IF
02,183 END IF
02,184 IF cryOLD <> cry THEN
02,185     UpdateCLine cryOLD, 1, NormalFRGr, NormalBCKGr, cryOLD
02,186     cryOLD = cry
02,187 ELSE 'it 'cry' could be changed by Mouse Wheel too, check it
02,188     AsIfItIsINKEY% = _MOUSEINPUT '    Check the mouse status
02,189     'WHILE _MOUSEINPUT: WEND '2021-Apr-23
02,190     IF AsIfItIsINKEY% THEN
02,191         buttondown1 = _MOUSEBUTTON(1)
02,192         buttondown2 = _MOUSEBUTTON(2)
02,193         buttondown3 = _MOUSEBUTTON(3)
02,194         mwheel = _MOUSEWHEEL
02,195     END IF
02,196
02,197     IF _KEYDOWN(LSHIFTkey&) OR _KEYDOWN(LCTRLkey&) OR _KEYDOWN(LALTkey&) THEN ShutDownIsNoLonger = 1 ' do not forget there are key+button shortcuts

```

```

02,198
02,199     IF (buttondown1 AND PrevClick1# = 0) THEN 'first detection of the click
02,200         PrevClick1# = TIMER(0.001)
02,201         SaveMouseX2 = _MOUSEX
02,202         SaveMouseY2 = _MOUSEY
02,203         ShutDownIsNoLonger = 0
02,204     END IF
02,205
02,206     IF (buttondown2 AND PrevClick2# = 0) THEN 'first detection of the click
02,207         PrevClick2# = TIMER(0.001)
02,208     END IF
02,209
02,210     IF (buttondown1 AND PrevClick1# <> 0) THEN 'already clicked
02,211         ForHowLongWasPressed1# = TIMER(0.001) - PrevClick1#
02,212         IF (SaveMouseX2 <> _MOUSEX) OR (SaveMouseY2 <> _MOUSEY) THEN ShutDownIsNoLonger = 1
02,213     END IF
02,214
02,215     IF (buttondown2 AND PrevClick2# <> 0) THEN 'already clicked
02,216         ForHowLongWasPressed2# = TIMER(0.001) - PrevClick2#
02,217     END IF
02,218
02,219     IF buttondown1 = 0 AND PrevClick1# <> 0 THEN 'write to the log
02,220         FOR MumboJumbo = 1 TO 3
02,221             Button1LOG_firstDetection#(MumboJumbo) = Button1LOG_firstDetection#(MumboJumbo + 1)
02,222             Button1LOG_ForHowLongHolded#(MumboJumbo) = Button1LOG_ForHowLongHolded#(MumboJumbo + 1)
02,223         NEXT
02,224         Button1LOG_firstDetection#(4) = PrevClick1#
02,225         Button1LOG_ForHowLongHolded#(4) = ForHowLongWasPressed1#
02,226         PrevClick1# = 0
02,227     END IF
02,228
02,229     IF buttondown2 = 0 AND PrevClick2# <> 0 THEN 'write to the log
02,230         FOR MumboJumbo = 1 TO 3
02,231             Button2LOG_firstDetection#(MumboJumbo) = Button2LOG_firstDetection#(MumboJumbo + 1)
02,232             Button2LOG_ForHowLongHolded#(MumboJumbo) = Button2LOG_ForHowLongHolded#(MumboJumbo + 1)
02,233         NEXT
02,234         Button2LOG_firstDetection#(4) = PrevClick2#
02,235         Button2LOG_ForHowLongHolded#(4) = ForHowLongWasPressed2#
02,236         PrevClick2# = 0
02,237     END IF
02,238
02,239     IF Button1LOG_firstDetection#(3) AND (Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(3) < DclickTime#) THEN Double_LeftClick = 1 ELSE Double_LeftClick = 0
02,240     'IF Double_LeftClick = 1 THEN PRINT "Double_LeftClick detected. Done in"; INT((Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(3)) * 1000); "ms."
02,241
02,242     IF Button2LOG_firstDetection#(3) AND (Button2LOG_firstDetection#(4) - Button2LOG_firstDetection#(3) < DclickTime#) THEN Double_RightClick = 1 ELSE Double_RightClick = 0
02,243     'IF Double_LeftClick = 1 THEN PRINT "Double_LeftClick detected. Done in"; INT((Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(3)) * 1000); "ms."
02,244
02,245     IF (ForHowLongWasPressed1# > ShutDownDuration#) AND (ShutDownIsNoLonger = 0) THEN 'the pointer should not be moved (in perfect case), inhere should be as the pointer during the initial click
02,246         ShutDown_LeftClick = 1: IF ShutDown_LeftClick = 1 THEN SYSTEM

```

```

02,247     END IF
02,248
02,249     IF _KEYDOWN(LSHIFTkey&) AND buttowndown1 THEN 'Fast Up
02,250         IF CurrentLine > 1 THEN CurrentLine = CurrentLine - 1
02,251         IF cry > 1 THEN
02,252             cry = cry - 1 'up
02,253         ELSE 'scrolling is needed
02,254             IF File_Frame_y > 1 THEN File_Frame_y = File_Frame_y - 1
02,255             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,256         END IF
02,257     END IF
02,258
02,259     IF _KEYDOWN(LSHIFTkey&) AND buttowndown2 THEN 'Fast Down
02,260         IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,261         IF cry < YdimROW THEN
02,262             IF cry < filecount THEN cry = cry + 1 'down
02,263         ELSE 'scrolling is needed
02,264             IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
02,265             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,266         END IF
02,267     END IF
02,268
02,269     IF _KEYDOWN(LALTkey&) AND buttowndown1 THEN 'LCTRL_HOME
02,270         LCTRL_HOME = 1
02,271     END IF
02,272     IF _KEYDOWN(LALTkey&) AND buttowndown2 THEN 'LCTRL_END
02,273         LCTRL_END = 1
02,274     END IF
02,275
02,276     IF _KEYDOWN(LCTRLkey&) AND buttowndown1 THEN 'Fast PgUp
02,277         IF File_Frame_y - YdimROW >= 1 THEN
02,278             File_Frame_y = File_Frame_y - YdimROW
02,279             IF CurrentLine - YdimROW >= 1 THEN CurrentLine = CurrentLine - YdimROW
02,280             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,281         END IF
02,282     END IF
02,283
02,284     IF _KEYDOWN(LCTRLkey&) AND buttowndown2 THEN 'Fast PgDn, grmb1, for some reason (LShift+button3) and (LShift+button1) are not spitting?!
02,285         IF File_Frame_y + YdimROW <= filecount - (YdimROW - 1) THEN
02,286             File_Frame_y = File_Frame_y + YdimROW
02,287             'IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,288             'IF CurrentLine + 1 <= filecount THEN CurrentLine = CurrentLine + 1
02,289             IF CurrentLine + YdimROW <= filecount THEN CurrentLine = CurrentLine + YdimROW
02,290             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,291         END IF
02,292     END IF
02,293
02,294     IF mwheel = 1 THEN ' as if Down
02,295         IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MzPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

02,296     IF cry < YdimROW THEN
02,297         IF cry < filecount THEN cry = cry + 1 'down
02,298     ELSE 'scrolling is needed
02,299         IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
02,300         UpdateWindowFrame NormalFBGr, NormalBCKGr
02,301     END IF
02,302 END IF
02,303 IF mwheel = -1 THEN ' as if Up
02,304     IF CurrentLine > 1 THEN CurrentLine = CurrentLine - 1
02,305     IF cry > 1 THEN
02,306         cry = cry - 1 'up
02,307     ELSE 'scrolling is needed
02,308         IF File_Frame_y > 1 THEN File_Frame_y = File_Frame_y - 1
02,309         UpdateWindowFrame NormalFBGr, NormalBCKGr
02,310     END IF
02,311 END IF
02,312 IF (_KEYDOWN(LSHIFTkey&) = 0) AND buttndown1 THEN
02,313     crxOLD = crx
02,314     cryOLD = cry
02,315     IF _MOUSEY <= MIN8(YdimROW, filecount) THEN
02,316         cry = _MOUSEY
02,317         crx = _MOUSEX
02,318         UpdateCLine cryOLD, 1, NormalFBGr, NormalBCKGr, cryOLD
02,319         DO WHILE cryOLD > cry
02,320             cryOLD = cryOLD - 1
02,321             IF CurrentLine > 1 THEN CurrentLine = CurrentLine - 1
02,322         LOOP
02,323         DO WHILE cryOLD < cry
02,324             cryOLD = cryOLD + 1
02,325             IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,326         LOOP
02,327     END IF
02,328 END IF
02,329
02,330 IF (mwheel = -1) AND buttndown2 THEN ' same as LCTRL_HOME
02,331     LCTRL_HOME = 1
02,332 END IF
02,333 IF (mwheel = 1) AND buttndown2 THEN ' same as LCTRL_END
02,334     LCTRL_END = 1
02,335 END IF
02,336
02,337 IF buttndown2 THEN ' 'drawing a line' gesture - 100 cells at least long, COMBO: Alt+X, Alt+Q
02,338     ReadOnceY = _MOUSEY
02,339     ReadOnceX = _MOUSEX
02,340     IF Button2Down = 0 THEN Mouse2Press! = TIMER: Button2Down = 1
02,341     IF WidenessINIT = 0 THEN WidenessINIT = ReadOnceX
02,342     IF HighnessINIT = 0 THEN HighnessINIT = ReadOnceY
02,343     IF ReadOnceY < YdimROW + 1 THEN LOCATE ReadOnceY, ReadOnceX: PRINT CHR$(176); 'don't write trail in status line
02,344 ELSE

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MzPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

02,345      Button2Down = 0
02,346      Mouse2Release! = TIMER
02,347      IF WidenessINIT AND (Mouse2Release! - Mouse2Press! > 0 AND Mouse2Release! - Mouse2Press! < 2) THEN
02,348          IF ABS(_MOUSEX - WidenessINIT) >= 100 THEN SYSTEM 'PLAY "v1018g" 'bidirectional i.e. can be drawn from left to right and vice versa
02,349          WidenessINIT = 0
02,350      END IF
02,351      'simulate PgDn (from bottom to top, as the hand cursor in PhotoShop), but not repetitive i.e. not-buffered
02,352      IF HighnessINIT AND (Mouse2Release! - Mouse2Press! > 0 AND Mouse2Release! - Mouse2Press! < 2) THEN
02,353          IF (HighnessINIT - _MOUSEY) >= 5 THEN 'PLAY "v1018g" 'bidirectional i.e. can be drawn from left to right and vice versa
02,354
02,355          IF File_Frame_y + YdimROW <= filecount - (YdimROW - 1) THEN
02,356              File_Frame_y = File_Frame_y + YdimROW
02,357              'IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,358              'IF CurrentLine + 1 <= filecount THEN CurrentLine = CurrentLine + 1
02,359              IF CurrentLine + YdimROW <= filecount THEN CurrentLine = CurrentLine + YdimROW
02,360              UpdateWindowFrame NormalFRGr, NormalBCKGr
02,361          END IF
02,362          ' Here is the layout:
02,363          ' The window-frame is 1..crx or 1..128 | The file-frame is 1..LongestLine
02,364          '
02,365          '
02,366          '          cry          60          |          filecount
02,367          '          FileArrayWINDOW$(60) |          FileArray(filecount)
02,368          ' if File_Frame_x < 128 then PADDING to 128 else File_Frame_x = 1..LongestLine-(128-1)
02,369          ' if File_Frame_y < 60 then PADDING to 60 else File_Frame_y = 1..filecount-(60-1)
02,370
02,371      END IF
02,372  END IF
02,373      'simulate PgUp (from top to bottom, as the hand cursor in PhotoShop), but not repetitive i.e. not-buffered
02,374      IF HighnessINIT AND (Mouse2Release! - Mouse2Press! > 0 AND Mouse2Release! - Mouse2Press! < 2) THEN
02,375          IF (_MOUSEY - HighnessINIT) >= 5 THEN 'PLAY "v1018g" 'bidirectional i.e. can be drawn from left to right and vice versa
02,376
02,377          IF File_Frame_y - YdimROW >= 1 THEN
02,378              File_Frame_y = File_Frame_y - YdimROW
02,379              IF CurrentLine - YdimROW >= 1 THEN CurrentLine = CurrentLine - YdimROW
02,380              UpdateWindowFrame NormalFRGr, NormalBCKGr
02,381          END IF
02,382
02,383      END IF
02,384  END IF
02,385      HighnessINIT = 0
02,386  END IF
02,387
02,388      IF LCTRL_RCTRL THEN PageScrollBenchmark = 1: TimeScrolla = TIMER
02,389      IF buttndown3 OR PageScrollBenchmark THEN 'PgDn - just add the page height i.e. 'YdimROW' to 'File_Frame_y'
02,390          ' Don't execute PgDn (advancing the 'CurrentLine') if 'File_Frame_y' is not "eligible":
02,391          'IF File_Frame_y < filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
02,392          'IF File_Frame_y + 1 <= filecount - (YdimROW - 1) THEN File_Frame_y = File_Frame_y + 1
02,393          'CAUTION [

```

```

02,394      ' Next three line beep NOT
02,395      'DEFLNG A-Z
02,396      'aaa = 61
02,397      'bbb = -48
02,398      'IF aaa <= bbb THEN BEEP: END
02,399      'Next three line beep
02,400      'aaa~& = 61
02,401      'bbb~& = -48
02,402      'IF aaa~& <= bbb~& THEN BEEP: END
02,403      'Next three line beep NOT
02,404      'aaa& = 61
02,405      'bbb& = -48
02,406      'IF aaa& <= bbb& THEN BEEP: END
02,407      'Next line works even when unsigned!
02,408      'IF File_Frame_y + YdimROW - (filecount - (YdimROW - 1)) <= 0 THEN
02,409      'CAUTION ]
02,410      'Next line doesn't work when unsigned!
02,411      IF File_Frame_y + YdimROW <= filecount - (YdimROW - 1) THEN
02,412          File_Frame_y = File_Frame_y + YdimROW
02,413          'IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,414          'IF CurrentLine + 1 <= filecount THEN CurrentLine = CurrentLine + 1
02,415          IF CurrentLine + YdimROW <= filecount THEN CurrentLine = CurrentLine + YdimROW
02,416          UpdateWindowFrame NormalFRGr, NormalBCKGr
02,417      ELSE
02,418          IF PageScrollBenchmark = 1 THEN
02,419              TimeScrollB = TIMER
02,420              _KEYCLEAR
02,421              ReportTimeToScroll MostRightField
02,422              PageScrollBenchmark = 0
02,423              _DISPLAY
02,424          END IF
02,425      END IF
02,426  END IF
02,427  END IF
02,428  IF cryOLD <> cry THEN
02,429      UpdateCLine cryOLD, 1, NormalFRGr, NormalBCKGr, cryOLD
02,430      cryOLD = cry
02,431  END IF
02,432  IF LCTRL_HOME THEN
02,433      LOCATE 1, 1, 1, 30, 31
02,434      crx = POS(0)
02,435      cry = CSRLIN
02,436      crxOLD = crx
02,437      cryOLD = cry
02,438      File_Frame_x = 1
02,439      File_Frame_y = 1
02,440      CurrentLine = 1
02,441      UpdateWindowFrame NormalFRGr, NormalBCKGr
02,442      'DO WHILE _MOUSEINPUT: LOOP

```

```

02,443     END IF
02,444     IF LCTRL_END THEN
02,445         IF filecount >= YdimROW THEN
02,446             IF filecount - (YdimROW - 1) THEN
02,447                 LOCATE YdimROW, 1, 1, 30, 31
02,448                 crx = POS(0)
02,449                 cry = CSRLIN
02,450                 crxOLD = crx
02,451                 cryOLD = cry
02,452                 File_Frame_x = 1
02,453                 File_Frame_y = filecount - (YdimROW - 1)
02,454                 CurrentLine = filecount
02,455                 UpdateWindowFrame NormalFRGr, NormalBCKGr
02,456             END IF
02,457         ELSE
02,458             LOCATE filecount, 1, 1, 30, 31
02,459             crx = POS(0)
02,460             cry = CSRLIN
02,461             crxOLD = crx
02,462             cryOLD = cry
02,463             File_Frame_x = 1
02,464             File_Frame_y = 1
02,465             CurrentLine = filecount
02,466             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,467         END IF
02,468         'DO WHILE _MOUSEINPUT: LOOP
02,469     END IF
02,470
02,471     ' @QB64team shared this clear code:
02,472     'DO
02,473     '    WHILE _MOUSEINPUT: WEND
02,474     '    IF _MOUSEBUTTON(1) THEN
02,475     '        IF timeElapsedSince!(lastClick!) <= .3 THEN
02,476     '            PRINT "... and make it double!"
02,477     '        ELSE
02,478     '            PRINT
02,479     '            PRINT "Click!";
02,480     '        END IF
02,481     '        lastClick! = TIMER
02,482     '        WHILE _MOUSEBUTTON(1): i = _MOUSEINPUT: WEND
02,483     '    END IF
02,484     '    _LIMIT 60
02,485     'LOOP UNTIL _KEYHIT = 27
02,486
02,487     'FUNCTION timeElapsedSince! (startTime!)
02,488     '    IF startTime! > TIMER THEN startTime! = startTime! - 86400
02,489     '    timeElapsedSince! = TIMER - startTime!
02,490     'END FUNCTION
02,491

```

```

02,492 'WHILE _MOUSEINPUT: WEND
02,493 'IF _MOUSEBUTTON(1) THEN
02,494 '    IF timeElapsedSince!(lastClick!) <= .3 THEN
02,495 '        ' Double-click
02,496 '        a$ = RTRIM$(FileArray$(CurrentLine))
02,497 '        IF _FILEEXISTS(a$) THEN
02,498 '            CLOSE #1
02,499 '            _MEMFREE MhandleOFF
02,500 '            _MEMFREE MhandleLEN
02,501 '            IF ToLoadOrNotFlag THEN
02,502 '                _MEMFREE Mwholefile
02,503 '            END IF
02,504 '            GOTO GettingStarted 'Before going above lines must be executed, i.e. to initialize.
02,505 '        ELSE ' not responsive enough - it misses some double-clicks?!
02,506 '            IF File_Frame_y - YdimROW >= 1 THEN
02,507 '                File_Frame_y = File_Frame_y - YdimROW
02,508 '                IF CurrentLine - YdimROW >= 1 THEN CurrentLine = CurrentLine - YdimROW
02,509 '                UpdateWindowFrame NormalFRGr, NormalBCKGr
02,510 '            END IF
02,511 '        END IF
02,512 '    ELSE
02,513 '        ' Single-click
02,514 '    END IF
02,515 '    lastClick! = TIMER
02,516 '    WHILE _MOUSEBUTTON(1): idummy = _MOUSEINPUT: WEND
02,517 'END IF
02,518
02,519 ' Grmbl, my doulbe-click sucks, therefore going to limbo... Yet, the above replacement nullifies/discards further events :( thus banning my other mouse functionality!
02,520 ' No grmbles anymore, fixed:
02,521 IF Double_LeftClick THEN
02,522     a$ = RTRIM$(FileArray$(CurrentLine))
02,523     IF _FILEEXISTS(a$) THEN
02,524         CLOSE #1
02,525         _MEMFREE MhandleOFF
02,526         _MEMFREE MhandleLEN
02,527         IF ToLoadOrNotFlag THEN
02,528             _MEMFREE Mwholefile
02,529         END IF
02,530         GOTO GettingStarted 'Before going above lines must be executed, i.e. to initialize.
02,531     ELSE ' not responsive enough - it misses some double-clicks?!
02,532         IF File_Frame_y - YdimROW >= 1 THEN
02,533             File_Frame_y = File_Frame_y - YdimROW
02,534             IF CurrentLine - YdimROW >= 1 THEN CurrentLine = CurrentLine - YdimROW
02,535             UpdateWindowFrame NormalFRGr, NormalBCKGr
02,536         END IF
02,537     END IF
02,538     Double_LeftClick = 0 'clear from buffer
02,539     FOR MumboJumbo = 1 TO 4
02,540         ButtonLOG_firstDetection#(MumboJumbo) = 0

```

```

02,541      Button1LOG_ForHowLongHolded#(MumboJumbo) = 0
02,542      NEXT
02,543  END IF
02,544
02,545  'WHILE _MOUSEINPUT: WEND
02,546  'IF _MOUSEBUTTON(2) THEN
02,547  '  IF timeElapsedSince!(lastClick!) <= .3 THEN
02,548  '    ' Double-click
02,549  '    IF File_Frame_y + YdimROW <= filecount - (YdimROW - 1) THEN
02,550  '      File_Frame_y = File_Frame_y + YdimROW
02,551  '      'IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,552  '      'IF CurrentLine + 1 <= filecount THEN CurrentLine = CurrentLine + 1
02,553  '      IF CurrentLine + YdimROW <= filecount THEN CurrentLine = CurrentLine + YdimROW
02,554  '      UpdateWindowFrame NormalFRGr, NormalBCKGr
02,555  '    END IF
02,556  '  ELSE
02,557  '    ' Single-click
02,558  '  END IF
02,559  '  lastClick! = TIMER
02,560  '  WHILE _MOUSEBUTTON(2): idummy = _MOUSEINPUT: WEND
02,561  'END IF
02,562
02,563  ' Grmb1, my doulbe-click sucks, therefore going to limbo... Yet, the above replacement nullifies/discards further events :( thus banning my other mouse functionality!
02,564  ' No grmb1s anymore, fixed:
02,565  IF Double_RightClick THEN
02,566  IF File_Frame_y + YdimROW <= filecount - (YdimROW - 1) THEN
02,567  File_Frame_y = File_Frame_y + YdimROW
02,568  'IF CurrentLine < filecount THEN CurrentLine = CurrentLine + 1
02,569  'IF CurrentLine + 1 <= filecount THEN CurrentLine = CurrentLine + 1
02,570  IF CurrentLine + YdimROW <= filecount THEN CurrentLine = CurrentLine + YdimROW
02,571  UpdateWindowFrame NormalFRGr, NormalBCKGr
02,572  END IF
02,573  Double_RightClick = 0 'clear from buffer
02,574  FOR MumboJumbo = 1 TO 4
02,575  Button2LOG_firstDetection#(MumboJumbo) = 0
02,576  Button2LOG_ForHowLongHolded#(MumboJumbo) = 0
02,577  NEXT
02,578  END IF
02,579
02,580  LOCATE cry, crx, 1, 30, 31
02,581  UpdateCLine cry, 1, InverseFRGr, InverseBCKGr, cry
02,582  UpdateCLL CurrentLine, LEN(Dumbo$) + 1 + 1
02,583  _DISPLAY
02,584
02,585  IF buttowndown2 AND buttowndown1 THEN 'pagoda, GRMBL, for now the file being handled SHOULD be in the invocation/home folder!
02,586  buttowndown2 = 0
02,587  buttowndown1 = 0 ' have to nullify because the cleaning of mouseinputs below prevents zeroing the variable in the main cycle where release is awaited but never registered.
02,588  ShutDownIsNoLonger = 1
02,589  UpdateCLineHIGHLIGHTword cry, 1, InverseFRGr, InverseBCKGr, cry

```

```

02,590      UpdateCLL CurrentLine, LEN(Dumbo$) + 1 + 1
02,591      LOCATE cry, crx, 0, 30, 31 'hide the cursor
02,592      _DISPLAY
02,593      IF LEN(CurrentWord$) > 1 THEN
02,594          UpdateNextToCCL_BUSY (MostRightField): _DISPLAY
02,595          p$ = LoadedFile$
02,596          IF INSTR(p$, Slash) THEN
02,597              pTRIMMED = 0
02,598              DO UNTIL MID$(p$, LEN(p$) - pTRIMMED, 1) = Slash
02,599                  pTRIMMED = pTRIMMED + 1
02,600              LOOP
02,601              p$ = RIGHT$(p$, pTRIMMED)
02,602          END IF
02,603          $IF WINDOWS THEN
02,604              $IF 32BIT THEN
02,605                  SHELL _HIDE "call " + "XGRAM_PAGODA5_32bit.bat" + " " + LCASE$(CurrentWord$) + " " + CHR$(34) + p$ + CHR$(34)
02,606              $END IF
02,607              $IF 64BIT THEN
02,608                  SHELL _HIDE "call " + "XGRAM_PAGODA5_64bit.bat" + " " + LCASE$(CurrentWord$) + " " + CHR$(34) + p$ + CHR$(34)
02,609              $END IF
02,610          $ELSE
02,611              SHELL _HIDE "sh " + "XGRAM_PAGODA5.sh" + " " + LCASE$(CurrentWord$) + " " + CHR$(34) + p$ + CHR$(34)
02,612          $END IF
02,613          UpdateNextToCCL_DONE (MostRightField): _DISPLAY
02,614          WHILE _MOUSEINPUT: WEND
02,615
02,616          LOCATE 1, 1, 0, 30, 31
02,617          FOR i = 1 TO YdimROW
02,618              LOCATE i, 1: PRINT SPACE$(XdimCOL);
02,619          NEXT
02,620
02,621          IF _FILEEXISTS(PSPlike$ + p$ + "_" + LCASE$(CurrentWord$) + ".PAGODA-order-5.txt") THEN
02,622              reportNAME$ = p$ + "_" + LCASE$(CurrentWord$) + ".PAGODA-order-5.txt"
02,623              pgd& = FREEFILE
02,624              OPEN PSPlike$ + p$ + "_" + LCASE$(CurrentWord$) + ".PAGODA-order-5.txt" FOR INPUT AS #pgd&
02,625              pgdlines = 0
02,626              DO WHILE NOT EOF(pgd&)
02,627                  pgdlines = pgdlines + 1
02,628                  LINE INPUT #pgd&, dummy1$
02,629              LOOP
02,630              CLOSE #pgd&
02,631              REDIM xgrams$(1 TO pgdlines)
02,632              SCROLLABLEheight = (YdimROW - 2) - 4 + 1
02,633              pgd& = FREEFILE
02,634              OPEN PSPlike$ + p$ + "_" + LCASE$(CurrentWord$) + ".PAGODA-order-5.txt" FOR INPUT AS #pgd&
02,635              LOCATE 1, 1
02,636              PRINT "PAGODA size: "; AddCommas$(pgdlines); " lines or "; AddCommas$(LOF(pgd&)); " bytes";
02,637              LOCATE 2, 1
02,638              PRINT "PAGODA name: "; LEFT$(reportNAME$, XdimCOL - LEN("PAGODA name: "));

```

```

02,639         pgdlines = 0
02,640         DO WHILE NOT EOF(pgd&)
02,641             pgdlines = pgdlines + 1
02,642             LINE INPUT #pgd&, xgrams$(pgdlines)
02,643             FOR j = 1 TO LEN(xgrams$(pgdlines))
02,644                 IF pgdlines MOD 2 = 0 THEN
02,645                     IF MID$(xgrams$(pgdlines), j, 1) = " " THEN MID$(xgrams$(pgdlines), j, 1) = "."
02,646                 END IF
02,647             NEXT
02,648         LOOP
02,649         CLOSE #pgd&
02,650         FOR i = 1 TO MIN$(SCROLLABLEheight, pgdlines) ' -2 due to PRINT: PRINT "Press ESC..."
02,651             LOCATE i + 3, 1: PRINT LEFT$(xgrams$(i), XdimCOL);
02,652         NEXT
02,653
02,654     END IF
02,655     PRINT
02,656     PRINT
02,657     PRINT "Press ESC... for vertical scroll use Up/Down, MouseLeftButton/MouseRightButton or Mouse Wheel.";
02,658     _DISPLAY
02,659     $IF WINDOWS THEN
02,660         PLAY "v20120g"
02,661     $END IF
02,662     LinesScroled = 0: LinesScroledMAX = pgdlines - SCROLLABLEheight
02,663     DO WHILE INKEY$ <> CHR$(27)
02,664         IF pgdlines > SCROLLABLEheight THEN
02,665             AsIfItIsINKEY% = _MOUSEINPUT ' Check the mouse status
02,666             buttondown1 = _MOUSEBUTTON(1)
02,667             buttondown2 = _MOUSEBUTTON(2)
02,668             buttondown3 = _MOUSEBUTTON(3)
02,669             mwheel = _MOUSEWHEEL
02,670             IF buttondown2 OR mwheel = 1 OR _KEYDOWN(DOWNkey&) THEN ' as if Down
02,671                 IF LinesScroled < LinesScroledMAX THEN
02,672                     LinesScroled = LinesScroled + 1
02,673                 END IF
02,674                 FOR i = 1 TO MIN$(SCROLLABLEheight, pgdlines) ' -2 due to PRINT: PRINT "Press ESC..."
02,675                     LOCATE i + 3, 1: PRINT CROPorPADatRIGHT$(xgrams$(i + LinesScroled), XdimCOL);
02,676                 NEXT
02,677             END IF
02,678             IF buttondown1 OR mwheel = -1 OR _KEYDOWN(UPkey&) THEN ' as if Up
02,679                 IF LinesScroled > 0 THEN
02,680                     LinesScroled = LinesScroled - 1
02,681                 END IF
02,682                 FOR i = 1 TO MIN$(SCROLLABLEheight, pgdlines) ' -2 due to PRINT: PRINT "Press ESC..."
02,683                     LOCATE i + 3, 1: PRINT CROPorPADatRIGHT$(xgrams$(i + LinesScroled), XdimCOL);
02,684                 NEXT
02,685             END IF
02,686             _LIMIT 500
02,687             _DISPLAY

```

```

02,688         END IF
02,689         LOOP
02,690         WHILE _MOUSEINPUT: WEND
02,691         UpdateWindowFrame NormalFRGr, NormalBCKGr
02,692     END IF
02,693 END IF
02,694
02,695 IF IsENTERreleased_Flag THEN ' Wrap the current/INVERSE line, and scroll it up/down as the PAGODA
02,696     Wwidth% = XdimCOL
02,697     UpdateCLL CurrentLine, LEN(Dumbo$) + 1 + 1
02,698     LOCATE cry, crx, 0, 30, 31 'hide the cursor
02,699     _DISPLAY
02,700     IF LEN(FileArrayFULL$(CurrentLine)) > XdimCOL THEN
02,701         UpdateNextToCCL_BUSY (MostRightField): _DISPLAY
02,702         l$ = FileArrayFULL$(CurrentLine)
02,703         ExpandTabsFULL l$
02,704         ' [[[
02,705         ' Firstly do the wrapping virtually (in order to avoid writing some wrapped chunks and encounter unwrappable chunk) - we need either a wrapped line (in its entirety) or none:
02,706         AssumeLineIsWrappable = 1
02,707         pgdlines = 0
02,708         lX$ = l$
02,709         DO WHILE LEN(lX$) > Wwidth%
02,710             Glupak% = Wwidth%
02,711             DO 'UNTIL (MID$(lX$, Glupak% + 1, 1) = " " OR MID$(lX$, Glupak% + 1, 1) = "\" OR MID$(lX$, Glupak% + 1, 1) = "/" OR MID$(lX$, Glupak% + 1, 1) = "_" OR MID$(lX$, Glupak%
+ 1, 1) = ",") AND MID$(lX$, Glupak%, 1) <> " "
02,712                 InvokeOnce$ = MID$(lX$, Glupak% + 1, 1)
02,713                 IF INSTR(" \_.,-!+%=:", InvokeOnce$) AND MID$(lX$, Glupak%, 1) <> " " THEN EXIT DO
02,714                 'IF (InvokeOnce$ = " " OR InvokeOnce$ = "\" OR InvokeOnce$ = "/" OR InvokeOnce$ = "_" OR InvokeOnce$ = "," OR InvokeOnce$ = "." OR InvokeOnce$ = "-" OR InvokeOnce$ = "!" OR
InvokeOnce$ = "+" OR InvokeOnce$ = "%") AND MID$(lX$, Glupak%, 1) <> " " THEN EXIT DO
02,715                 Glupak% = Glupak% + 1
02,716                 IF Glupak% = 0 THEN
02,717                     AssumeLineIsWrappable = 0
02,718                     pgdlines = 0
02,719                     GOTO B4TxpanarVIRTUAL2
02,720                 END IF
02,721             LOOP
02,722             lX$ = LTRIM$(MID$(lX$, Glupak% + 1, LEN(lX$) - (Glupak%)))
02,723             pgdlines = pgdlines + 1
02,724         LOOP
02,725         pgdlines = pgdlines + 1 ' don't forget the last chunk/remainder
02,726         REDIM xgrams$(1 TO pgdlines)
02,727
02,728         B4TxpanarVIRTUAL2:
02,729         pgdlines = 0
02,730         IF AssumeLineIsWrappable = 1 THEN
02,731             lX$ = l$
02,732             DO WHILE LEN(lX$) > Wwidth%
02,733                 Glupak% = Wwidth%
02,734                 DO 'UNTIL (MID$(lX$, Glupak% + 1, 1) = " " OR MID$(lX$, Glupak% + 1, 1) = "\" OR MID$(lX$, Glupak% + 1, 1) = "/" OR MID$(lX$, Glupak% + 1, 1) = "_" OR MID$(lX$, Glupak% + 1, 1) = "," OR MID$(lX$,

```

```

Glupak% + 1, 1) = ",") AND MID$(1X$, Glupak%, 1) <> " "
02,735         InvokeOnce$ = MID$(1X$, Glupak% + 1, 1)
02,736         IF INSTR(" \/_;.-!+%=:", InvokeOnce$) AND MID$(1X$, Glupak%, 1) <> " " THEN EXIT DO
02,737         'IF (InvokeOnce$ = " " OR InvokeOnce$ = "\" OR InvokeOnce$ = "/" OR InvokeOnce$ = "_" OR InvokeOnce$ = ";" OR InvokeOnce$ = "," OR InvokeOnce$ = "." OR InvokeOnce$ = "-" OR InvokeOnce$ = "!" OR
InvokeOnce$ = "+" OR InvokeOnce$ = "%") AND MID$(1X$, Glupak%, 1) <> " " THEN EXIT DO
02,738         Glupak% = Glupak% - 1
02,739         IF Glupak% = 0 THEN
02,740             GOTO B4Txpanar2
02,741         END IF
02,742     LOOP
02,743     pgdlines = pgdlines + 1
02,744     xgrams$(pgdlines) = LEFT$(1X$, Glupak%) 'PRINT #3, LEFT$(1X$, Glupak%)
02,745     1X$ = LTRIM$(MID$(1X$, Glupak% + 1, LEN(1X$) - (Glupak%)))
02,746     LOOP
02,747     pgdlines = pgdlines + 1 ' don't forget the last chunk/remainder
02,748     xgrams$(pgdlines) = 1X$ 'PRINT #3, 1X$
02,749     B4Txpanar2:
02,750 ELSE 'unwrappable
02,751     $IF WINDOWS THEN
02,752         PLAY "v20160aba"
02,753     $END IF
02,754 END IF
02,755
02,756 ']]]
02,757 UpdateNextToCCL_DONE (MostRightField): _DISPLAY
02,758 WHILE _MOUSEINPUT: WEND
02,759
02,760 IF pgdlines THEN
02,761     LOCATE 1, 1, 0, 30, 31
02,762     FOR i = 1 TO YdimROW
02,763         LOCATE i, 1: PRINT SPACE$(XdimCOL);
02,764     NEXT
02,765
02,766     SCROLLABLEheight = (YdimROW - 2) - 3 + 1
02,767
02,768     LOCATE 1, 1, 0
02,769     PRINT "Original Current/INVERSE line size: "; AddCommas$(LEN(FileArrayFULL$(CurrentLine))); " bytes ";
02,770
02,771     FOR i = 1 TO MIN$(SCROLLABLEheight, pgdlines) ' -2 due to PRINT: PRINT "Press ESC..."
02,772         LOCATE i + 2, 1: PRINT LEFT$(xgrams$(i), XdimCOL);
02,773     NEXT
02,774
02,775     PRINT
02,776     PRINT
02,777     PRINT "Release Enter (or press Esc)... for vertical scroll use Up/Down, MouseLeftButton/MouseRightButton or Mouse Wheel.";
02,778     _DISPLAY
02,779
02,780     $IF WINDOWS THEN
02,781         PLAY "v20120g"

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

02,782 $END IF
02,783 LinesScroled = 0: LinesScroledMAX = pgdlines - SCROLLABLEheight
02,784 DO WHILE INKEY$ <> CHR$(27) AND IsENTERreleased = 0
02,785     IF pgdlines > SCROLLABLEheight THEN
02,786         AsIfItIsINKEY% = _MOUSEINPUT '      Check the mouse status
02,787         buttowndown1 = _MOUSEBUTTON(1)
02,788         buttowndown2 = _MOUSEBUTTON(2)
02,789         buttowndown3 = _MOUSEBUTTON(3)
02,790         mwheel = _MOUSEWHEEL
02,791         IF buttowndown2 OR mwheel = 1 OR _KEYDOWN(DOWNkey&) THEN ' as if Down
02,792             IF LinesScroled < LinesScroledMAX THEN
02,793                 LinesScroled = LinesScroled + 1
02,794             END IF
02,795             FOR i = 1 TO MIN(SCROLLABLEheight, pgdlines) ' -2 due to PRINT: PRINT "Press ESC..."
02,796                 LOCATE i + 2, 1: PRINT CROPorPADatRIGHT$(xgrams$(i + LinesScroled), XdimCOL);
02,797             NEXT
02,798         END IF
02,799         IF buttowndown1 OR mwheel = -1 OR _KEYDOWN(UPkey&) THEN ' as if Up
02,800             IF LinesScroled > 0 THEN
02,801                 LinesScroled = LinesScroled - 1
02,802             END IF
02,803             FOR i = 1 TO MIN(SCROLLABLEheight, pgdlines) ' -2 due to PRINT: PRINT "Press ESC..."
02,804                 LOCATE i + 2, 1: PRINT CROPorPADatRIGHT$(xgrams$(i + LinesScroled), XdimCOL);
02,805             NEXT
02,806         END IF
02,807         _LIMIT 500
02,808         _DISPLAY
02,809     END IF
02,810     LOOP
02,811     WHILE _MOUSEINPUT: WEND
02,812     UpdateWindowFrame NormalFRGr, NormalBCKGr
02,813 END IF 'IF pgdlines THEN
02,814 END IF
02,815 _KEYCLEAR 2 'in order to get rid of all eventual 'Enter' hits during above scrolling
02,816 END IF 'IF IsENTERreleased_Flag THEN
02,817
02,818 'DO WHILE INKEY$ <> "": LOOP ' have to clear the keyboard buffer
02,819 _LIMIT 500 ' _LIMIT 30 'commented because the wheel up/down was not working?!
02,820 'Caution: Above line works fine with "mainstream" mouse like a cheap HP 3-button one, but when attaching "game" mouse as White Shark Lancelot, it became superlaggy and unresponsive, maybe due to REPORT RATE aka POLL
RATE being high, had to set it to 125Hz in order to work as if "cheap".
02,821 LOOP
02,822
02,823 END
02,824
02,825 SUB ReturnCOMBO
02,826     SHARED LSHIFT_RSHIFT
02,827     SHARED LCTRL_RCTRL
02,828     SHARED LALT_RALT
02,829

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

02,830 SHARED LALT_HOME
02,831 SHARED LALT_END
02,832
02,833 SHARED LALT_INS
02,834 SHARED LALT_DEL
02,835
02,836 SHARED LALT_PGUP
02,837 SHARED LALT_PGDN
02,838
02,839 SHARED LALT_Left
02,840 SHARED LALT_Right
02,841 SHARED LALT_Up
02,842 SHARED LALT_Down
02,843
02,844 SHARED RALT_HOME
02,845 SHARED RALT_END
02,846
02,847 SHARED RALT_INS
02,848 SHARED RALT_DEL
02,849
02,850 SHARED RALT_PGUP
02,851 SHARED RALT_PGDN
02,852
02,853 SHARED RALT_Left
02,854 SHARED RALT_Right
02,855 SHARED RALT_Up
02,856 SHARED RALT_Down
02,857
02,858 SHARED LSHIFT_HOME
02,859 SHARED LSHIFT_END
02,860
02,861 SHARED LSHIFT_INS
02,862 SHARED LSHIFT_DEL
02,863
02,864 SHARED LSHIFT_PGUP
02,865 SHARED LSHIFT_PGDN
02,866
02,867 SHARED LSHIFT_Left
02,868 SHARED LSHIFT_Right
02,869 SHARED LSHIFT_Up
02,870 SHARED LSHIFT_Down
02,871
02,872 SHARED RSHIFT_HOME
02,873 SHARED RSHIFT_END
02,874
02,875 SHARED RSHIFT_INS
02,876 SHARED RSHIFT_DEL
02,877
02,878 SHARED RSHIFT_PGUP

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

02,879 SHARED RSHIFT_PGDN
02,880
02,881 SHARED RSHIFT_Left
02,882 SHARED RSHIFT_Right
02,883 SHARED RSHIFT_Up
02,884 SHARED RSHIFT_Down
02,885
02,886 SHARED LCTRL_HOME
02,887 SHARED LCTRL_END
02,888
02,889 SHARED LCTRL_INS
02,890 SHARED LCTRL_DEL
02,891
02,892 SHARED LCTRL_PGUP
02,893 SHARED LCTRL_PGDN
02,894
02,895 SHARED LCTRL_Left
02,896 SHARED LCTRL_Right
02,897 SHARED LCTRL_Up
02,898 SHARED LCTRL_Down
02,899
02,900 SHARED RCTRL_HOME
02,901 SHARED RCTRL_END
02,902
02,903 SHARED RCTRL_INS
02,904 SHARED RCTRL_DEL
02,905
02,906 SHARED RCTRL_PGUP
02,907 SHARED RCTRL_PGDN
02,908
02,909 SHARED RCTRL_Left
02,910 SHARED RCTRL_Right
02,911 SHARED RCTRL_Up
02,912 SHARED RCTRL_Down
02,913
02,914 SHARED LSHIFT_LCTRL_HOME
02,915 SHARED LSHIFT_LCTRL_END
02,916
02,917 SHARED LSHIFT_LCTRL_INS
02,918 SHARED LSHIFT_LCTRL_DEL
02,919
02,920 SHARED LSHIFT_LCTRL_PGUP
02,921 SHARED LSHIFT_LCTRL_PGDN
02,922
02,923 SHARED LSHIFT_LCTRL_Left
02,924 SHARED LSHIFT_LCTRL_Right
02,925 SHARED LSHIFT_LCTRL_Up
02,926 SHARED LSHIFT_LCTRL_Down
02,927

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MzPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

02,928  SHARED LSHIFT_BackSpace
02,929  SHARED LSHIFT_TAB
02,930  SHARED LSHIFT_SPACE
02,931  SHARED LSHIFT_ESC
02,932  SHARED LSHIFT_ENTER
02,933
02,934  SHARED RSHIFT_BackSpace
02,935  SHARED RSHIFT_TAB
02,936  SHARED RSHIFT_SPACE
02,937  SHARED RSHIFT_ESC
02,938  SHARED RSHIFT_ENTER
02,939
02,940  SHARED LCTRL_SPACE
02,941  SHARED LCTRL_ENTER
02,942
02,943  SHARED RCTRL_SPACE
02,944  SHARED RCTRL_ENTER
02,945
02,946  IF _KEYDOWN(LSHIFTkey%) AND _KEYDOWN(RSHIFTkey%) THEN LSHIFT_RSHIFT = 1 ELSE LSHIFT_RSHIFT = 0
02,947  IF _KEYDOWN(LCTRLkey%) AND _KEYDOWN(RCTRLkey%) THEN LCTRL_RCTRL = 1 ELSE LCTRL_RCTRL = 0
02,948  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(RALTkey%) THEN LALT_RALT = 1 ELSE LALT_RALT = 0
02,949
02,950  'LALT:
02,951  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(HOMEkey%) THEN LALT_HOME = 1 ELSE LALT_HOME = 0
02,952  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(ENDkey%) THEN LALT_END = 1 ELSE LALT_END = 0
02,953
02,954  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(INSkey%) THEN LALT_INS = 1 ELSE LALT_INS = 0
02,955  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(DELkey%) THEN LALT_DEL = 1 ELSE LALT_DEL = 0
02,956
02,957  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(PGUPkey%) THEN LALT_PGUP = 1 ELSE LALT_PGUP = 0
02,958  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(PGDNkey%) THEN LALT_PGDN = 1 ELSE LALT_PGDN = 0
02,959
02,960  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(LEFTkey%) THEN LALT_Left = 1 ELSE LALT_Left = 0
02,961  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(RIGHTkey%) THEN LALT_Right = 1 ELSE LALT_Right = 0
02,962  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(UPkey%) THEN LALT_Up = 1 ELSE LALT_Up = 0
02,963  IF _KEYDOWN(LALTkey%) AND _KEYDOWN(DOWNkey%) THEN LALT_Down = 1 ELSE LALT_Down = 0
02,964
02,965  'RALT:
02,966  IF _KEYDOWN(RALTkey%) AND _KEYDOWN(HOMEkey%) THEN RALT_HOME = 1 ELSE RALT_HOME = 0
02,967  IF _KEYDOWN(RALTkey%) AND _KEYDOWN(ENDkey%) THEN RALT_END = 1 ELSE RALT_END = 0
02,968
02,969  IF _KEYDOWN(RALTkey%) AND _KEYDOWN(INSkey%) THEN RALT_INS = 1 ELSE RALT_INS = 0
02,970  IF _KEYDOWN(RALTkey%) AND _KEYDOWN(DELkey%) THEN RALT_DEL = 1 ELSE RALT_DEL = 0
02,971
02,972  IF _KEYDOWN(RALTkey%) AND _KEYDOWN(PGUPkey%) THEN RALT_PGUP = 1 ELSE RALT_PGUP = 0
02,973  IF _KEYDOWN(RALTkey%) AND _KEYDOWN(PGDNkey%) THEN RALT_PGDN = 1 ELSE RALT_PGDN = 0
02,974
02,975  IF _KEYDOWN(RALTkey%) AND _KEYDOWN(LEFTkey%) THEN PRINT RALT_Left = 1 ELSE RALT_Left = 0
02,976  IF _KEYDOWN(RALTkey%) AND _KEYDOWN(RIGHTkey%) THEN PRINT RALT_Right = 1 ELSE RALT_Right = 0

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MzPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

02,977 IF _KEYDOWN(RALTkey&) AND _KEYDOWN(UPkey&) THEN RALT_Up = 1 ELSE RALT_Up = 0
02,978 IF _KEYDOWN(RALTkey&) AND _KEYDOWN(DOWNkey&) THEN RALT_Down = 1 ELSE RALT_Down = 0
02,979
02,980 'LSHIFT:
02,981 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(HOMEkey&) THEN LSHIFT_HOME = 1 ELSE LSHIFT_HOME = 0
02,982 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(ENDkey&) THEN LSHIFT_END = 1 ELSE LSHIFT_END = 0
02,983
02,984 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(INSkey&) THEN LSHIFT_INS = 1 ELSE LSHIFT_INS = 0
02,985 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(DELkey&) THEN LSHIFT_DEL = 1 ELSE LSHIFT_DEL = 0
02,986
02,987 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(PGUPkey&) THEN LSHIFT_PGUP = 1 ELSE LSHIFT_PGUP = 0
02,988 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(PGDNkey&) THEN LSHIFT_PGDN = 1 ELSE LSHIFT_PGDN = 0
02,989
02,990 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LEFTkey&) THEN LSHIFT_Left = 1 ELSE LSHIFT_Left = 0
02,991 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(RIGHTkey&) THEN LSHIFT_Right = 1 ELSE LSHIFT_Right = 0
02,992 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(UPkey&) THEN LSHIFT_Up = 1 ELSE LSHIFT_Up = 0
02,993 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(DOWNkey&) THEN LSHIFT_Down = 1 ELSE LSHIFT_Down = 0
02,994
02,995 'RSHIFT:
02,996 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(HOMEkey&) THEN RSHIFT_HOME = 1 ELSE RSHIFT_HOME = 0
02,997 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(ENDkey&) THEN RSHIFT_END = 1 ELSE RSHIFT_END = 0
02,998
02,999 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(INSkey&) THEN RSHIFT_INS = 1 ELSE RSHIFT_INS = 0
03,000 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(DELkey&) THEN RSHIFT_DEL = 1 ELSE RSHIFT_DEL = 0
03,001
03,002 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(PGUPkey&) THEN RSHIFT_PGUP = 1 ELSE RSHIFT_PGUP = 0
03,003 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(PGDNkey&) THEN RSHIFT_PGDN = 1 ELSE RSHIFT_PGDN = 0
03,004
03,005 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(LEFTkey&) THEN RSHIFT_Left = 1 ELSE RSHIFT_Left = 0
03,006 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(RIGHTkey&) THEN RSHIFT_Right = 1 ELSE RSHIFT_Right = 0
03,007 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(UPkey&) THEN RSHIFT_Up = 1 ELSE RSHIFT_Up = 0
03,008 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(DOWNkey&) THEN RSHIFT_Down = 1 ELSE RSHIFT_Down = 0
03,009
03,010 'LCTRL:
03,011 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(HOMEkey&) THEN LCTRL_HOME = 1 ELSE LCTRL_HOME = 0
03,012 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(ENDkey&) THEN LCTRL_END = 1 ELSE LCTRL_END = 0
03,013
03,014 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(INSkey&) THEN LCTRL_INS = 1 ELSE LCTRL_INS = 0
03,015 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(DELkey&) THEN LCTRL_DEL = 1 ELSE LCTRL_DEL = 0
03,016
03,017 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(PGUPkey&) THEN LCTRL_PGUP = 1 ELSE LCTRL_PGUP = 0
03,018 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(PGDNkey&) THEN LCTRL_PGDN = 1 ELSE LCTRL_PGDN = 0
03,019
03,020 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(LEFTkey&) THEN LCTRL_Left = 1 ELSE LCTRL_Left = 0
03,021 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(RIGHTkey&) THEN LCTRL_Right = 1 ELSE LCTRL_Right = 0
03,022 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(UPkey&) THEN LCTRL_Up = 1 ELSE LCTRL_Up = 0
03,023 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(DOWNkey&) THEN LCTRL_Down = 1 ELSE LCTRL_Down = 0
03,024
03,025 'RCTRL:

```

```

03,026 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(HOMEkey&) THEN RCTRL_HOME = 1 ELSE RCTRL_HOME = 0
03,027 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(ENDkey&) THEN RCTRL_END = 1 ELSE RCTRL_END = 0
03,028
03,029 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(INSkey&) THEN RCTRL_INS = 1 ELSE RCTRL_INS = 0
03,030 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(DELkey&) THEN RCTRL_DEL = 1 ELSE RCTRL_DEL = 0
03,031
03,032 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(PGUPkey&) THEN RCTRL_PGUP = 1 ELSE RCTRL_PGUP = 0
03,033 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(PGDNkey&) THEN RCTRL_PGDN = 1 ELSE RCTRL_PGDN = 0
03,034
03,035 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(LEFTkey&) THEN RCTRL_Left = 1 ELSE RCTRL_Left = 0
03,036 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(RIGHTkey&) THEN RCTRL_Right = 1 ELSE RCTRL_Right = 0
03,037 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(UPkey&) THEN RCTRL_Up = 1 ELSE RCTRL_Up = 0
03,038 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(DOWNkey&) THEN RCTRL_Down = 1 ELSE RCTRL_Down = 0
03,039
03,040 'LSHIFT+LCTRL: NOTICE: LSHIFT+LCTRL+Left triggers 3 variables on - 1] LSHIFT_LCTRL_Left 2] LSHIFT_Left 3] LCTRL_Left
03,041 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(HOMEkey&) THEN LSHIFT_LCTRL_HOME = 1 ELSE LSHIFT_LCTRL_HOME = 0
03,042 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(ENDkey&) THEN LSHIFT_LCTRL_END = 1 ELSE LSHIFT_LCTRL_END = 0
03,043
03,044 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(INSkey&) THEN LSHIFT_LCTRL_INS = 1 ELSE LSHIFT_LCTRL_INS = 0
03,045 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(DELkey&) THEN LSHIFT_LCTRL_DEL = 1 ELSE LSHIFT_LCTRL_DEL = 0
03,046
03,047 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(PGUPkey&) THEN LSHIFT_LCTRL_PGUP = 1 ELSE LSHIFT_LCTRL_PGUP = 0
03,048 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(PGDNkey&) THEN LSHIFT_LCTRL_PGDN = 1 ELSE LSHIFT_LCTRL_PGDN = 0
03,049
03,050 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(LEFTkey&) THEN LSHIFT_LCTRL_Left = 1 ELSE LSHIFT_LCTRL_Left = 0
03,051 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(RIGHTkey&) THEN LSHIFT_LCTRL_Right = 1 ELSE LSHIFT_LCTRL_Right = 0
03,052 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(UPkey&) THEN LSHIFT_LCTRL_Up = 1 ELSE LSHIFT_LCTRL_Up = 0
03,053 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(LCTRLkey&) AND _KEYDOWN(DOWNkey&) THEN LSHIFT_LCTRL_Down = 1 ELSE LSHIFT_LCTRL_Down = 0
03,054
03,055 'LSHIFT:
03,056 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(BACKSPCkey&) THEN LSHIFT_BackSpace = 1 ELSE LSHIFT_BackSpace = 0
03,057 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(TABkey&) THEN LSHIFT_TAB = 1 ELSE LSHIFT_TAB = 0
03,058 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(SPACEkey&) THEN LSHIFT_SPACE = 1 ELSE LSHIFT_SPACE = 0
03,059 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(ESCkey&) THEN LSHIFT_ESC = 1 ELSE LSHIFT_ESC = 0
03,060 IF _KEYDOWN(LSHIFTkey&) AND _KEYDOWN(ENTERkey&) THEN LSHIFT_ENTER = 1 ELSE LSHIFT_ENTER = 0
03,061 'RSHIFT:
03,062 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(BACKSPCkey&) THEN RSHIFT_BackSpace = 1 ELSE RSHIFT_BackSpace = 0
03,063 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(TABkey&) THEN RSHIFT_TAB = 1 ELSE RSHIFT_TAB = 0
03,064 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(SPACEkey&) THEN RSHIFT_SPACE = 1 ELSE RSHIFT_SPACE = 0
03,065 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(ESCkey&) THEN RSHIFT_ESC = 1 ELSE RSHIFT_ESC = 0
03,066 IF _KEYDOWN(RSHIFTkey&) AND _KEYDOWN(ENTERkey&) THEN RSHIFT_ENTER = 1 ELSE RSHIFT_ENTER = 0
03,067
03,068 'LCTRL:
03,069 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(SPACEkey&) THEN LCTRL_SPACE = 1 ELSE LCTRL_SPACE = 0
03,070 IF _KEYDOWN(LCTRLkey&) AND _KEYDOWN(ENTERkey&) THEN LCTRL_ENTER = 1 ELSE LCTRL_ENTER = 0
03,071 'RCTRL:
03,072 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(SPACEkey&) THEN RCTRL_SPACE = 1 ELSE RCTRL_SPACE = 0
03,073 IF _KEYDOWN(RCTRLkey&) AND _KEYDOWN(ENTERkey&) THEN RCTRL_ENTER = 1 ELSE RCTRL_ENTER = 0
03,074

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,075 ' _KEYCLEAR
03,076 END SUB
03,077
03,078 SUB ExpandTabsFULL (l$)
03,079     SHARED XdimCOL
03,080     IF INSTR(l$, CHR$(9)) THEN
03,081         TabV% = 8
03,082         b$ = "": f& = 0
03,083         PossibleRecalculation = LEN(l$)
03,084         'FOR i& = 1 TO MIN&(PossibleRecalculation, XdimCOL) 'LEN(l$) ' For some reason going all the length is too slow...?! As it is, the lateral scroll is crippled!
03,085         FOR i& = 1 TO PossibleRecalculation
03,086             RemoveSpecial$ = MID$(l$, i&, 1)
03,087             IF RemoveSpecial$ = CHR$(9) THEN
03,088                 b$ = b$ + STRING$((f& \ TabV%) * TabV% + TabV% - f&, " ")
03,089                 ' |
03,090                 ' |
03,091                 ' \|/
03,092                 'TabV% - (f% - (f% \ TabV%) * TabV%) =
03,093                 'TabV% - (f% MOD TabV%)
03,094                 f& = (f& \ TabV%) * TabV% + TabV%
03,095             ELSE
03,096                 b$ = b$ + RemoveSpecial$ 'MID$(l$, i&, 1)
03,097                 f& = f& + 1
03,098             END IF
03,099         NEXT
03,100         l$ = b$
03,101     END IF
03,102 END SUB
03,103
03,104 SUB ExpandTabs (l$)
03,105     SHARED XdimCOL
03,106     IF INSTR(l$, CHR$(9)) THEN
03,107         TabV% = 8
03,108         b$ = "": f& = 0
03,109         PossibleRecalculation = LEN(l$)
03,110         'FOR i& = 1 TO MIN&(PossibleRecalculation, XdimCOL) 'LEN(l$) ' For some reason going all the length is too slow...?! As it is, the lateral scroll is crippled!
03,111         RemoveSpecial$ = MID$(l$, i&, 1)
03,112         IF RemoveSpecial$ = CHR$(9) THEN
03,113             b$ = b$ + STRING$((f& \ TabV%) * TabV% + TabV% - f&, " ")
03,114             ' |
03,115             ' |
03,116             ' \|/
03,117             'TabV% - (f% - (f% \ TabV%) * TabV%) =
03,118             'TabV% - (f% MOD TabV%)
03,119             f& = (f& \ TabV%) * TabV% + TabV%
03,120         ELSE
03,121             b$ = b$ + RemoveSpecial$ 'MID$(l$, i&, 1)
03,122             f& = f& + 1
03,123         END IF

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

03,124      NEXT
03,125      l$ = b$
03,126      END IF
03,127 END SUB
03,128
03,129 FUNCTION MIN& (YdimROW, filecount)
03,130     IF YdimROW < filecount THEN MIN& = YdimROW ELSE MIN& = filecount
03,131 END FUNCTION
03,132
03,133 FUNCTION AddCommas$ (numeral)
03,134     s$ = LTRIM$(STR$(numeral))
03,135     IF LEN(s$) > 3 THEN
03,136         IF (LEN(s$) MOD 3) THEN x$ = STRING$(3 - (LEN(s$) MOD 3), " ") + s$ ELSE x$ = s$
03,137         s$ = ""
03,138         FOR i = 1 TO LEN(x$) STEP 3
03,139             s$ = s$ + MID$(x$, i, 3) + ", "
03,140         NEXT
03,141         s$ = LEFT$(s$, LEN(s$) - 1)
03,142     END IF
03,143     AddCommas$ = LTRIM$(s$)
03,144 END FUNCTION
03,145
03,146 FUNCTION AddCommasPaddedNUM$ (numeral)
03,147     SHARED FieldLineNum
03,148     s$ = LTRIM$(STR$(numeral))
03,149     IF LEN(s$) > 3 THEN
03,150         IF (LEN(s$) MOD 3) THEN x$ = STRING$(3 - (LEN(s$) MOD 3), " ") + s$ ELSE x$ = s$
03,151         s$ = ""
03,152         FOR i = 1 TO LEN(x$) STEP 3
03,153             s$ = s$ + MID$(x$, i, 3) + ", "
03,154         NEXT
03,155         s$ = LEFT$(s$, LEN(s$) - 1)
03,156     END IF
03,157     s$ = LTRIM$(s$)
03,158     Padded$ = RIGHT$("000,000,000,000", FieldLineNum)
03,159     MID$(Padded$, LEN(Padded$) - LEN(s$) + 1, LEN(s$)) = s$
03,160     AddCommasPaddedNUM$ = Padded$
03,161 END FUNCTION
03,162
03,163 FUNCTION AddCommasPaddedLEN$ (numeral)
03,164     SHARED FieldLineLen
03,165     s$ = LTRIM$(STR$(numeral))
03,166     IF LEN(s$) > 3 THEN
03,167         IF (LEN(s$) MOD 3) THEN x$ = STRING$(3 - (LEN(s$) MOD 3), " ") + s$ ELSE x$ = s$
03,168         s$ = ""
03,169         FOR i = 1 TO LEN(x$) STEP 3
03,170             s$ = s$ + MID$(x$, i, 3) + ", "
03,171         NEXT
03,172         s$ = LEFT$(s$, LEN(s$) - 1)

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

03,173     END IF
03,174     s$ = LTRIM$(s$)
03,175     Padded$ = RIGHT$("000,000,000,000", FieldLineLen)
03,176     MID$(Padded$, LEN(Padded$) - LEN(s$) + 1, LEN(s$)) = s$
03,177     AddCommasPaddedLEN$ = Padded$
03,178 END FUNCTION
03,179
03,180 SUB UpdateCLL (1, posit)
03,181     SHARED YdimROW ' , FileArray$( )
03,182     crx = POS(0)
03,183     cry = CSRLIN
03,184     LOCATE YdimROW + 1, posit: COLOR 9, 0
03,185     'PRINT "; Line Number: " + AddCommasPaddedNUM$(1);
03,186     PRINT "; Line: " + AddCommasPaddedNUM$(1);
03,187     'PRINT "; Line Length: " + AddCommasPaddedLEN$(LEN(FileArrayFULL$(1))); 'fix with 'FULL' from jun-02
03,188     PRINT "; Linesize: " + AddCommasPaddedLEN$(LEN(FileArrayFULL$(1))); 'fix with 'FULL' from jun-02
03,189     LOCATE cry, crx, 1, 30, 31
03,190 END SUB
03,191
03,192 SUB UpdateNextToCCL_BUSY (posit)
03,193     SHARED XdimCOL
03,194     SHARED YdimROW
03,195     crx = POS(0)
03,196     cry = CSRLIN
03,197     LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT SPACE$((XdimCOL - posit) - 0);
03,198     LOCATE YdimROW + 1, posit: COLOR 9, 0: PRINT "; Status: BUSY";
03,199     LOCATE cry, crx, 1, 30, 31
03,200 END SUB
03,201
03,202 SUB UpdateNextToCCL_DONE (posit)
03,203     SHARED XdimCOL
03,204     SHARED YdimROW
03,205     crx = POS(0)
03,206     cry = CSRLIN
03,207     LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT SPACE$((XdimCOL - posit) - 0); 'STRING$((XdimCOL - posit) - 1, " ");
03,208     LOCATE YdimROW + 1, posit: COLOR 9, 0: PRINT "; Status: DONE";
03,209     LOCATE cry, crx, 1, 30, 31
03,210 END SUB
03,211
03,212 SUB UpdateNextToCCL_UNWRAPPABLEcount (posit)
03,213     SHARED XdimCOL
03,214     SHARED YdimROW
03,215     SHARED UnwrappableLines
03,216     crx = POS(0)
03,217     cry = CSRLIN
03,218     LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT SPACE$((XdimCOL - posit) - 0);
03,219     LOCATE YdimROW + 1, posit: COLOR 9, 0: PRINT "; Unwrappable Lines: " + AddCommas$(UnwrappableLines);
03,220     LOCATE cry, crx, 1, 30, 31
03,221 END SUB

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

03,222
03,223 SUB UpdateCLine (lineToWrite, columnToWrite, FRGR, BACKGR, ln)
03,224     SHARED FileArrayWINDOW$( )
03,225     SHARED XdimCOL
03,226     SHARED CurrentWord$
03,227     crx = POS(0)
03,228     cry = CSRLIN
03,229     COLOR FRGR, BACKGR
03,230     LOCATE lineToWrite, columnToWrite, 1, 30, 31
03,231     PRINT FileArrayWINDOW$(ln);
03,232     'get the word under the cursor [
03,233     LeftM = crx: RightM = crx 'both can go within 1..XdimCOL
03,234     CurrentChar$ = MID$(FileArrayWINDOW$(ln), LeftM, 1)
03,235     IF (CurrentChar$ >= "a" AND CurrentChar$ <= "z") OR (CurrentChar$ >= "A" AND CurrentChar$ <= "Z") THEN
03,236         DO WHILE LeftM > 1
03,237             CurrentChar$ = MID$(FileArrayWINDOW$(ln), LeftM - 1, 1)
03,238             IF (CurrentChar$ >= "a" AND CurrentChar$ <= "z") OR (CurrentChar$ >= "A" AND CurrentChar$ <= "Z") THEN
03,239                 LeftM = LeftM - 1
03,240             ELSE
03,241                 EXIT DO
03,242             END IF
03,243         LOOP
03,244         DO WHILE RightM < XdimCOL
03,245             CurrentChar$ = MID$(FileArrayWINDOW$(ln), RightM + 1, 1)
03,246             IF (CurrentChar$ >= "a" AND CurrentChar$ <= "z") OR (CurrentChar$ >= "A" AND CurrentChar$ <= "Z") THEN
03,247                 RightM = RightM + 1
03,248             ELSE
03,249                 EXIT DO
03,250             END IF
03,251         LOOP
03,252     END IF
03,253     IF RightM - LeftM > 0 THEN CurrentWord$ = MID$(FileArrayWINDOW$(ln), LeftM, RightM - LeftM + 1) ELSE CurrentWord$ = ""
03,254     'get the word under the cursor ]
03,255     LOCATE cry, crx, 1, 30, 31
03,256 END SUB
03,257
03,258 SUB UpdateCLineHIGHLIGHTword (lineToWrite, columnToWrite, FRGR, BACKGR, ln)
03,259     SHARED FileArrayWINDOW$( )
03,260     SHARED XdimCOL
03,261     SHARED CurrentWord$
03,262     crx = POS(0)
03,263     cry = CSRLIN
03,264     COLOR FRGR, BACKGR
03,265     LOCATE lineToWrite, columnToWrite, 1, 30, 31
03,266     PRINT FileArrayWINDOW$(ln);
03,267     'get the word under the cursor [
03,268     LeftM = crx: RightM = crx 'both can go within 1..XdimCOL
03,269     CurrentChar$ = MID$(FileArrayWINDOW$(ln), LeftM, 1)
03,270     IF (CurrentChar$ >= "a" AND CurrentChar$ <= "z") OR (CurrentChar$ >= "A" AND CurrentChar$ <= "Z") THEN

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,271 DO WHILE LeftM > 1
03,272     CurrentChar$ = MID$(FileArrayWINDOW$(ln), LeftM - 1, 1)
03,273     IF (CurrentChar$ >= "a" AND CurrentChar$ <= "z") OR (CurrentChar$ >= "A" AND CurrentChar$ <= "Z") THEN
03,274         LeftM = LeftM - 1
03,275     ELSE
03,276         EXIT DO
03,277     END IF
03,278 LOOP
03,279 DO WHILE RightM < XdimCOL
03,280     CurrentChar$ = MID$(FileArrayWINDOW$(ln), RightM + 1, 1)
03,281     IF (CurrentChar$ >= "a" AND CurrentChar$ <= "z") OR (CurrentChar$ >= "A" AND CurrentChar$ <= "Z") THEN
03,282         RightM = RightM + 1
03,283     ELSE
03,284         EXIT DO
03,285     END IF
03,286 LOOP
03,287 END IF
03,288 IF RightM - LeftM > 0 THEN CurrentWord$ = MID$(FileArrayWINDOW$(ln), LeftM, RightM - LeftM + 1) ELSE CurrentWord$ = ""
03,289 IF CurrentWord$ <> "" THEN
03,290     COLOR 7 + 8, BACKGR
03,291     FOR qq = LeftM TO RightM
03,292         LOCATE cry, qq, 1, 30, 31
03,293         PRINT MID$(FileArrayWINDOW$(ln), qq, 1);
03,294     NEXT
03,295 END IF
03,296 'get the word under the cursor ]
03,297 LOCATE cry, crx, 1, 30, 31
03,298 END SUB
03,299
03,300 SUB UpdateWindowFrame (FRGB, BACKGR)
03,301     SHARED YdimROW, filecount, File_Frame_y, File_Frame_x, XdimCOL, FileArrayWINDOW$(), FileArray$()
03,302     COLOR FRGB, BACKGR
03,303     IF YdimROW > filecount THEN
03,304         FOR i = 1 TO YdimROW
03,305             LOCATE i, 1: PRINT SPACE$(XdimCOL);
03,306         NEXT
03,307     END IF
03,308     FOR i = 1 TO MIN$(YdimROW, filecount)
03,309         DumboReadOnceNotThrice$ = FileArray$(i + (File_Frame_y - 1))
03,310         IF LEN(DumboReadOnceNotThrice$) >= XdimCOL THEN
03,311             FileArrayWINDOW$(i) = MID$(DumboReadOnceNotThrice$, File_Frame_x, XdimCOL)
03,312         ELSE
03,313             FileArrayWINDOW$(i) = DumboReadOnceNotThrice$ + SPACE$(XdimCOL - LEN(DumboReadOnceNotThrice$))
03,314         END IF
03,315         'since r.8 the goal is to browse properly binary/.tar files
03,316         'FOR j = 1 TO LEN(FileArrayWINDOW$(i))
03,317         'since r.9 the goal is to browse properly German/Spanish/French/Italian files, SO COMMENTING OUT:
03,318         'IF MID$(FileArrayWINDOW$(i), j, 1) < " " THEN MID$(FileArrayWINDOW$(i), j, 1) = CHR$(32)
03,319         'NEXT

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

03,320      LOCATE i, 1: PRINT FileArrayWINDOW$(i);
03,321      NEXT
03,322 END SUB
03,323
03,324 SUB ReportTimeToLoad (posit)
03,325     SHARED YdimROW, XdimCOL
03,326     SHARED TimeA, TimeB
03,327     crx = POS(0)
03,328     cry = CSRLIN
03,329     ElapsedTime% = TimeB - TimeA
03,330     IF ElapsedTime% < 0 THEN ElapsedTime% = (86400 - TimeA) + TimeB
03,331     Paddedstr$ = "; Loaded in " + LTRIM$(STR$(ElapsedTime%)) + " seconds."
03,332     '      P=6
03,333     '[123456] (8-P)-1=2 ' 2021-Jul-26, ugh, the fix is without '-1' because P is not 5 but 6!
03,334     LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT SPACE$((XdimCOL - posit) - 0);
03,335     LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT Paddedstr$;
03,336     LOCATE cry, crx, 1, 30, 31
03,337 END SUB
03,338
03,339 SUB ReportTimeToScroll (posit)
03,340     SHARED YdimROW, XdimCOL
03,341     SHARED TimeScrollA, TimeScrollB
03,342     SHARED LineScrollBenchmark
03,343     SHARED PageScrollBenchmark
03,344     SHARED filecount
03,345     crx = POS(0)
03,346     cry = CSRLIN
03,347     ElapsedTime% = TimeScrollB - TimeScrollA
03,348     IF ElapsedTime% < 0 THEN ElapsedTime% = (86400 - TimeScrollA) + TimeScrollB
03,349     IF ElapsedTime% = 0 THEN ElapsedTime% = 1
03,350     'Line-By-Line or Line granularity Scroll Rate; LPS stands for Lines-Per-Second
03,351     'Page-By-Page or Page granularity Scroll Rate; PPS stands for Pages-Per-Second
03,352     LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT SPACE$((XdimCOL - posit) - 0);
03,353     IF XdimCOL = 198 THEN
03,354         IF LineScrollBenchmark = 1 THEN LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT "; DOWNed in"; ElapsedTime%; "seconds, or "; AddCommasPaddedLEN$(filecount / ElapsedTime%); " LPS";
03,355         IF PageScrollBenchmark = 1 THEN LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT "; PGDNed in"; ElapsedTime%; "seconds, or "; AddCommasPaddedLEN$((filecount / YdimROW) / ElapsedTime%); " PPS";
03,356     ELSE
03,357         IF LineScrollBenchmark = 1 THEN LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT "; DOWNed in"; ElapsedTime%; "seconds.";
03,358         IF PageScrollBenchmark = 1 THEN LOCATE YdimROW + 1, posit: COLOR 4, 0: PRINT "; PGDNed in"; ElapsedTime%; "seconds.";
03,359     END IF
03,360     LOCATE cry, crx, 1, 30, 31
03,361 END SUB
03,362
03,363 FUNCTION FileArrayFULL$(i)
03,364     SHARED QWORD, LineLen13
03,365     'SHARED TheWholeFile$
03,366     _MEMGET MhandleOFF, MhandleOFF.OFFSET + 8&& * (i - 1), QWORD
03,367     _MEMGET MhandleLEN, MhandleLEN.OFFSET + 8&& * (i - 1), LineLen13
03,368     IF ToLoadOrNotFlag THEN

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,369      'BufferForLine$ = MID$(TheWholeFile$, QWORD, LineLen13)
03,370      BufferForLine$ = SPACE$(LineLen13)
03,371      _MEMGET Mwholefile, Mwholefile.OFFSET + (QWORD - 1), BufferForLine$
03,372      ELSE
03,373      BufferForLine$ = SPACE$(LineLen13)
03,374      SEEK #1, QWORD
03,375      GET #1, , BufferForLine$
03,376      END IF
03,377      'ExpandTabs (BufferForLine$)
03,378      FileArrayFULL$ = BufferForLine$
03,379 END FUNCTION
03,380
03,381 FUNCTION FileArray$ (i)
03,382     SHARED QWORD, LineLen13
03,383     'SHARED TheWholeFile$
03,384     _MEMGET MhandleOFF, MhandleOFF.OFFSET + 8&& * (i - 1), QWORD
03,385     _MEMGET MhandleLEN, MhandleLEN.OFFSET + 8&& * (i - 1), LineLen13
03,386     IF ToLoadOrNotFlag THEN
03,387         'BufferForLine$ = MID$(TheWholeFile$, QWORD, LineLen13)
03,388         BufferForLine$ = SPACE$(LineLen13)
03,389         _MEMGET Mwholefile, Mwholefile.OFFSET + (QWORD - 1), BufferForLine$
03,390     ELSE
03,391         BufferForLine$ = SPACE$(LineLen13)
03,392         SEEK #1, QWORD
03,393         GET #1, , BufferForLine$
03,394     END IF
03,395     ExpandTabs (BufferForLine$)
03,396     FileArray$ = BufferForLine$
03,397 END FUNCTION
03,398
03,399 SUB ShowIndigoWindow
03,400     'COLOR 8, 0: LOCATE i, 1 + LEN(FileArrayWINDOW$(i)): PRINT CHR$(179);: COLOR FRGR, BACKGR
03,401 END SUB
03,402
03,403 'http://rosettacode.org/wiki/Levenshtein_distance#FutureBasic
03,404 'FutureBasic
03,405 'Based on Wikipedia algorithm. Suitable for Pascal strings.
03,406
03,407 'include "ConsoleWindow"
03,408
03,409 'local fn LevenshteinDistance( aStr as Str255, bStr as Str255 ) as long
03,410 'dim as long m, n, i, j, min, k, l
03,411 'dim as long distance( 255, 255 )
03,412
03,413 'm = len(aStr)
03,414 'n = len(bStr)
03,415
03,416 'for i = 0 to m
03,417     distance( i, 0 ) = i

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,418 'next
03,419
03,420 'for j = 0 to n
03,421 '    distance( 0, j ) = j
03,422 'next
03,423
03,424 'for j = 1 to n
03,425 '    for i = 1 to m
03,426 '        if mid$( aStr, i, 1 ) == mid$( bStr, j, 1 )
03,427 '            distance( i, j ) = distance( i-1, j-1 )
03,428 '        else
03,429 '            min = distance( i-1, j ) + 1
03,430 '            k   = distance( i, j - 1 ) + 1
03,431 '            l   = distance( i-1, j-1 ) + 1
03,432 '            if k < min then min = k
03,433 '            if l < min then min = l
03,434 '            distance( i, j ) = min
03,435 '        end if
03,436 '    next
03,437 'next
03,438 'end fn = distance( m, n )
03,439
03,440 'dim as long i
03,441 'dim as Str255 testStr( 5, 2 )
03,442
03,443 'testStr( 0, 0 ) = "kitten"      : testStr( 0, 1 ) = "sitting"
03,444 'testStr( 1, 0 ) = "rosettacode" : testStr( 1, 1 ) = "raisethysword"
03,445 'testStr( 2, 0 ) = "Saturday"    : testStr( 2, 1 ) = "Sunday"
03,446 'testStr( 3, 0 ) = "FutureBasic" : testStr( 3, 1 ) = "FutureBasic"
03,447 'testStr( 4, 0 ) = "here's a bunch of words"
03,448 'testStr( 4, 1 ) = "to wring out this code"
03,449
03,450 'for i = 0 to 4
03,451 '    print "1st string = "; testStr( i, 0 )
03,452 '    print "2nd string = "; testStr( i, 1 )
03,453 '    print "Levenshtein distance ="; fn LevenshteinDistance( testStr( i, 0 ), testStr( i, 1 ) )
03,454 '    print
03,455 'next
03,456
03,457 'Output:
03,458
03,459 '1st string = kitten
03,460 '2nd string = sitting
03,461 'Levenshtein distance = 3
03,462
03,463 '1st string = rosettacode
03,464 '2nd string = raisethysword
03,465 'Levenshtein distance = 8
03,466

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,467 '1st string = Saturday
03,468 '2nd string = Sunday
03,469 'Levenshtein distance = 3
03,470
03,471 '1st string = FutureBasic
03,472 '2nd string = FutureBasic
03,473 'Levenshtein distance = 0
03,474
03,475 '1st string = here's a bunch of words
03,476 '2nd string = to wring out this code
03,477 'Levenshtein distance = 18
03,478
03,479 'In information theory and computer science, the Levenshtein distance is a metric for measuring the amount of difference between two sequences (i.e. an edit distance). The Levenshtein distance between two strings is
defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character.
03,480
03,481 SUB DrawBoxShadow3 (TopLrow, TopLcol, BottomRow, BottomRcol, Captmn$)
03,482     SHARED FileSize
03,483     SHARED LoadedFile$
03,484     SHARED PSPlike$
03,485     SHARED YdimROW, filecount, File_Frame_y, XdimCOL, FileArrayWINDOW$(), FileArray$()
03,486     'Shadow
03,487     BckGRcolor = 4
03,488     COLOR 8, BACKGR
03,489     FOR i = 1 TO MIN8(YdimROW, filecount)
03,490         'DumboReadOnceNotThrice$ = FileArray$(i + (File_Frame_y - 1))
03,491         'IF LEN(DumboReadOnceNotThrice$) >= XdimCOL THEN
03,492             '    FileArrayWINDOW$(i) = MID$(DumboReadOnceNotThrice$, 1, XdimCOL)
03,493         'ELSE
03,494             '    FileArrayWINDOW$(i) = DumboReadOnceNotThrice$ + SPACE$(XdimCOL - LEN(DumboReadOnceNotThrice$))
03,495         'END IF
03,496         IF i >= TopLrow + 1 AND i <= BottomRow + 1 THEN
03,497             LOCATE i, TopLcol + 1: PRINT MID$(FileArrayWINDOW$(i), TopLcol + 1, BottomRcol - TopLcol + 1);
03,498         END IF
03,499     NEXT
03,500     'Outer frame
03,501     LOCATE TopLrow, TopLcol
03,502     COLOR 7, BckGRcolor
03,503     PRINT CHR$(218); STRING$(BottomRcol - TopLcol - 1, CHR$(196));
03,504     COLOR 0, BckGRcolor: PRINT CHR$(191);
03,505     COLOR 7, BckGRcolor: LOCATE TopLrow, TopLcol + 2: PRINT "[ ";: COLOR 15, BckGRcolor: PRINT Captmn$;: COLOR 7, BckGRcolor: PRINT " ]";
03,506     FOR i = TopLrow + 1 TO BottomRow - 1
03,507         COLOR 7, BckGRcolor: LOCATE i, TopLcol: PRINT CHR$(179); STRING$(BottomRcol - TopLcol - 1, CHR$(32));: COLOR 0, BckGRcolor: PRINT CHR$(179);
03,508     NEXT
03,509     LOCATE BottomRow, TopLcol
03,510     COLOR 7, BckGRcolor: PRINT CHR$(192);
03,511     COLOR 0, BckGRcolor: PRINT STRING$(BottomRcol - TopLcol - 1, CHR$(196)); CHR$(217)
03,512     'Inner frame
03,513     'In case of YdimROW = 40 then shrunk panel
03,514     IF YdimROW = 40 THEN shrunkP = 14 ELSE shrunkP = 0

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

03,564 LOCATE , TopLcol + 2
03,565 PRINT "      wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty}."
03,566 LOCATE , TopLcol + 2
03,567 PRINT "Note3: Two FAST wildcards are available:"
03,568 LOCATE , TopLcol + 2
03,569 PRINT "      wildcard '&' any character(s) or empty,"
03,570 LOCATE , TopLcol + 2
03,571 PRINT "      wildcard '+' any character and not empty."
03,572 LOCATE , TopLcol + 2
03,573 PRINT "Note4: Don't mix SLOW and FAST, the SLOW overrides the FAST. Also, 1 byte exact pattern, not allowed."
03,574 IF shrunkP = 0 THEN
03,575     LOCATE , TopLcol + 2
03,576     PRINT "Exact Case-Sensitive 16-threaded search for all lines up to 26208 chars long:"
03,577     LOCATE , TopLcol + 2
03,578     PRINT "Example1: Arnold"
03,579     LOCATE , TopLcol + 2
03,580     PRINT "Example2: "; CHR$(34); "metal fatigue"; CHR$(34)
03,581     LOCATE , TopLcol + 2
03,582     PRINT "Wildcard Case-Insensitive 16-threaded search for all lines up to 26208 chars long:"
03,583     LOCATE , TopLcol + 2
03,584     PRINT "Example3: "; CHR$(34); "out~~~~~ize*"; CHR$(34)
03,585     LOCATE , TopLcol + 2
03,586     PRINT "      Possible hits within a line starting with O/o: outhyperbolize, OUTSIZE, outsized"
03,587     LOCATE , TopLcol + 2
03,588     PRINT "NORMAL/EXHAUSTIVE Fuzzy Case-Insensitive 16-threaded search for all lines up to 156/26208 chars long:"
03,589     LOCATE , TopLcol + 2
03,590     PRINT "Example4: 3 psychedlicize"
03,591     LOCATE , TopLcol + 2
03,592     PRINT "      This line '1234psychedlicize' won't match."
03,593     LOCATE , TopLcol + 2
03,594     PRINT "Example5: 2e edelvais"
03,595     LOCATE , TopLcol + 2
03,596     PRINT "      Possible hits: edelweiss, edelweisses, psychedelicism"
03,597     LOCATE , TopLcol + 2
03,598     PRINT "Note5: Levenshtein search can be NORMAL, as in Example4, matching the whole line."
03,599     LOCATE , TopLcol + 2
03,600     PRINT "Note6: Levenshtein search can be EXHAUSTIVE, as in Example5, if LD is postfixed with 'e',"
03,601     LOCATE , TopLcol + 2
03,602     PRINT "      matching each position in the line."
03,603     LOCATE , TopLcol + 2
03,604     PRINT "Note7: The Needle/Pattern below is set by default as the Current_Word or the CLIPBOARD."
03,605 END IF
03,606 COLOR 7, BckGColor
03,607 HelpLine$ = "Edit keys, allowed: Home, Left, Right, End, Backspace, Del, Esc"
03,608 LOCATE TopLrow + 41 - shrunkP, TopLcol + 2: PRINT HelpLine$
03,609 _DISPLAY
03,610 AddToKazahana$ = InputLine$(TopLrow + 39 - shrunkP, TopLcol + 2, BottomRcol - TopLcol - 1 - 2)
03,611 StartSt$ = DATE$ + " " + TIME$
03,612 LOCATE TopLrow + 41 - shrunkP, TopLcol + 2: PRINT STRING$(LEN(HelpLine$), " ")

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,613 LOCATE TopRow + 41 - shrunkP, TopLcol + 2: PRINT "Start: "; StartSt$
03,614 LOCATE 1, 1, 0, 30, 31
03,615 _DISPLAY
03,616
03,617 TimeK1# = TIMER(.01)
03,618 $IF WINDOWS THEN
03,619     PRINT #13, CHR$(34) + PSPlike$ + "Kazahana.exe" + CHR$(34) + " " + AddToKazahana$ + " " + CHR$(34) + LoadedFile$ + CHR$(34) + " 30539"
03,620 $END IF
03,621 $IF WINDOWS THEN
03,622     'SHELL _DONTWAIT _HIDE CHR$(34) + PSPlike$ + "Kazahana_Hexadecad_GCC_472_SSE41_32bit.exe" + CHR$(34) + " " + AddToKazahana$ + " " + CHR$(34) + LoadedFile$ + CHR$(34) + " 1539" '9 is odd therefore no dump of
pattern field, 9 is bigger than 4 therefore case insensitive wildcard search.
03,623     SHELL _HIDE CHR$(34) + PSPlike$ + "Kazahana.exe" + CHR$(34) + " " + AddToKazahana$ + " " + CHR$(34) + LoadedFile$ + CHR$(34) + " 30539" '9 is odd therefore no dump of pattern field, 9 is bigger than 4 therefore
case insensitive wildcard search.
03,624 $ELSE
03,625     Shell _Hide Chr$(34) + PSPlike$ + "kazahana" + Chr$(34) + " " + AddToKazahana$ + " " + Chr$(34) + LoadedFile$ + Chr$(34) + " 30539" '9 is odd therefore no dump of pattern field, 9 is bigger than 4 therefore case
insensitive wildcard search.
03,626 $END IF
03,627 TimeK2# = TIMER(.01)
03,628 SearchTime# = TimeK2# - TimeK1#
03,629 IF SearchTime# < 0 THEN SearchTime# = (86400 - TimeK1#) + TimeK2#
03,630 LOCATE TopRow + 42 - shrunkP, TopLcol + 2: PRINT "Ready: "; DATE$ + " " + TIME$
03,631 f = FREEFILE
03,632 OPEN "Kazahana.txt" FOR BINARY AS #f
03,633 KazLen = LOF(f)
03,634 CLOSE #f
03,635 SearchRate = FileSize / (SearchTime# + 0.0001)
03,636 LOCATE TopRow + 43 - shrunkP, TopLcol + 2: PRINT "Size of Kazahana.txt: "; AddCommas$(KazLen); " bytes"; "; Search-and-Dump Speed: "; AddCommas$(SearchRate); " bytes/s";
03,637 LOCATE TopRow + 44 - shrunkP, TopLcol + 2: PRINT "Press Esc...";
03,638 _DISPLAY
03,639 $IF WINDOWS THEN
03,640     PLAY "v20120g"
03,641 $END IF
03,642 DO WHILE INKEY$ <> CHR$(27)
03,643     _LIMIT 30
03,644 LOOP
03,645 LOCATE 1, 1, 1, 30, 31
03,646
03,647 'E:\_KAZE_Smxt_Benchmarks\QB64_kit_v1.4.2_48 GB\qb64\Kazahana_Hexadecad_GCC_472_SSE41_32bit.exe
03,648 'Kazahana, a typhoon-class exact & wildcards & Levenshtein Distance (Wagner-Fischer) searcher, r. 1---fix+nowait_critical_nixFIX_Wolfram+fixITER+EX+CS_fix_DEFINE_Trolldom, copyleft Kaze 2019-May-21.
03,649 'Usage: Kazahana [AtMostLevenshteinDistance][e] string textualfile MasterBufferSize
03,650 'Note0: MasterBufferSize is in KB, consider 1024, 3072, 7168 or bigger (up to 2GB). Three additional flags were mapped on this value: all dump
03,651 '         lines (except fuzzy's) will have/lack pattern-source info when the number is even/odd respectively, see Examples #5 and #6.
03,652 '         When MasterBufferSize ends in 0, then No-Dump i.e. hits are only counted.
03,653 '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.
03,654 'Note1: There are three regimes: exact, wildcards and fuzzy searches. First two kick in when 3 parameters are given, fuzzy when 4.
03,655 'Note2: What decides whether exact or wildcards? Of course presence of at least one wildcard. To see exact search see Example #4.
03,656 'Note3: Exact search hits with 'Railgun_Trolldom', not 'Railgun_Sekireigan_Wolfram'.
03,657 'Note4a: Incoming string is automatically lowercased for fuzzy searches i.e. they are case insensitive.
03,658 'Note4b: Incoming string is NOT automatically lowercased for wildcards searches when MasterBufferSize ends in 0.4 i.e. they are case sensitive.

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

03,659 'Note4c: Incoming string is automatically lowercased for wildcards searches when MasterBufferSize ends in 5..9 i.e. they are case insensitive.
03,660 'Note5: Incoming string could be up to 26208/156 chars for Exact&Wildcard&ExhaustiveFuzzy/Fuzzy respectively.
03,661 'Note5a: Since 2013-Nov-21 Levenshtein search exits not when the incoming line is bigger than 156 chars, now it just skips longer lines.
03,662 'Note5b: Since 2013-Dec-05 Levenshtein search can be EXHAUSTIVE if LD is postfixed with 'e'.
03,663 'Note6: Incoming textuallfile could be bigger than 4GB.
03,664 'Note7: Each line should end with [CR]LF, that is Windows or/and UNIX style.
03,665 'Note8: The dump goes to Kazahana.txt file.
03,666 'Note9a: Nine SLOW wildcards are available:
03,667 '   wildcard '*' any character(s) or empty,
03,668 '   wildcard '.' any ALPHA character(s) or empty,
03,669 '   wildcard '"' any NON-ALPHA character(s) or empty,
03,670 '   wildcard '@'/'#' any character {or empty}/{and not empty},
03,671 '   wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
03,672 '   wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty}.
03,673 'Note9b: Two FAST wildcards are available:
03,674 '   wildcard '&' any character(s) or empty,
03,675 '   wildcard '+' any character and not empty.
03,676 '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.
03,677 'Example1: E:\Kazahana 0 ramjet MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536
03,678 'Example2: E:\Kazahana 3 psychedlicize MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536
03,679 'Example3: E:\Kazahana "psyched~~~~~ize^" MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536
03,680 'Example4: E:\Kazahana "metal fatigue" enwiki-20121201-pages-articles.xml 7168
03,681 'Example5: E:\Kazahana "out~~~~~ize*" MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536
03,682 '   E:\>type Kazahana.txt
03,683 '   [out~~~~~ize*] outhyperbolize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
03,684 '   [out~~~~~ize*] outsize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
03,685 '   [out~~~~~ize*] outsized /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
03,686 '   [out~~~~~ize*] outstrategize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
03,687 '   [out~~~~~ize*] outtyrannize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
03,688 'Example6: E:\Kazahana "out~~~~~ize*" MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1537
03,689 '   E:\>type Kazahana.txt
03,690 '   outhyperbolize
03,691 '   outsize
03,692 '   outsized
03,693 '   outstrategize
03,694 '   outtyrannize
03,695 'Example7: E:\Kazahana 2e edelvais MASAKARI_General-Purpose_Grade_English_Wordlist.wrd 1024
03,696 '   E:\>type Kazahana.txt
03,697 '   bordelais
03,698 '   bordelaise
03,699 '   edelweiss
03,700 '   edelweisses
03,701 '   foredevised
03,702 '   predellas
03,703 '   psychedelicism
03,704 'Info1: One second seems to have 1,000 clocks.
03,705
03,706 'E:\_KAZE_Smxrt_Benchmarks\QB64_kit_v1.4_2.48_GB\qb64>
03,707 END SUB

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

03,708
03,709 FUNCTION DrawBoxShadow3exact$ (TopLrow, TopLcol, BottomRow, BottomRcol, Captm$)
03,710     SHARED FileSize
03,711     SHARED LoadedFile$
03,712     SHARED PSPLike$
03,713     SHARED YdimROW, filecount, File_Frame_y, XdimCOL, FileArrayWINDOW$( ), FileArray$( )
03,714     'Shadow
03,715     BckGRcolor = 6
03,716     COLOR 8, BckGR
03,717     FOR i = 1 TO MIN$(YdimROW, filecount)
03,718         'DumboReadOnceNotThrice$ = FileArray$(i + (File_Frame_y - 1))
03,719         'IF LEN(DumboReadOnceNotThrice$) >= XdimCOL THEN
03,720             '    FileArrayWINDOW$(i) = MID$(DumboReadOnceNotThrice$, 1, XdimCOL)
03,721         'ELSE
03,722             '    FileArrayWINDOW$(i) = DumboReadOnceNotThrice$ + SPACE$(XdimCOL - LEN(DumboReadOnceNotThrice$))
03,723         'END IF
03,724         IF i >= TopLrow + 1 AND i <= BottomRow + 1 THEN
03,725             LOCATE i, TopLcol + 1: PRINT MID$(FileArrayWINDOW$(i), TopLcol + 1, BottomRcol - TopLcol + 1);
03,726         END IF
03,727     NEXT
03,728     'Outer frame
03,729     LOCATE TopLrow, TopLcol
03,730     COLOR 7, BckGRcolor
03,731     PRINT CHR$(218); STRING$(BottomRcol - TopLcol - 1, CHR$(196));
03,732     COLOR 0, BckGRcolor: PRINT CHR$(191);
03,733     COLOR 7, BckGRcolor: LOCATE TopLrow, TopLcol + 2: PRINT "[ ";: COLOR 15, BckGRcolor: PRINT Captm$;: COLOR 7, BckGRcolor: PRINT " ]";
03,734     FOR i = TopLrow + 1 TO BottomRow - 1
03,735         COLOR 7, BckGRcolor: LOCATE i, TopLcol: PRINT CHR$(179); STRING$(BottomRcol - TopLcol - 1, CHR$(32));: COLOR 0, BckGRcolor: PRINT CHR$(179);
03,736     NEXT
03,737     LOCATE BottomRow, TopLcol
03,738     COLOR 7, BckGRcolor: PRINT CHR$(192);
03,739     COLOR 0, BckGRcolor: PRINT STRING$(BottomRcol - TopLcol - 1, CHR$(196)); CHR$(217)
03,740     'Inner frame
03,741     'In case of YdimROW = 40 then shrunk panel
03,742     shrunkP = 14
03,743     LOCATE TopLrow + 38 - 23 - shrunkP, TopLcol + 1
03,744     COLOR 0, BckGRcolor
03,745     PRINT CHR$(218); STRING$(BottomRcol - 1 - (TopLcol + 1) - 1, CHR$(196));
03,746     COLOR 7, BckGRcolor: PRINT CHR$(191);
03,747     FOR i = TopLrow + 39 - 23 - shrunkP TO TopLrow + 40 - 23 - shrunkP
03,748         COLOR 0, BckGRcolor: LOCATE i, TopLcol + 1: PRINT CHR$(179); STRING$(BottomRcol - 1 - (TopLcol + 1) - 1, CHR$(32));: COLOR 7, BckGRcolor: PRINT CHR$(179);
03,749     NEXT
03,750     LOCATE TopLrow + 40 - 23 - shrunkP, TopLcol + 1
03,751     COLOR 0, BckGRcolor: PRINT CHR$(192);
03,752     COLOR 7, BckGRcolor: PRINT STRING$(BottomRcol - 1 - (TopLcol + 1) - 1, CHR$(196)); CHR$(217)
03,753
03,754     LOCATE TopLrow + 2, TopLcol + 2
03,755     COLOR 7, BckGRcolor
03,756

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,757 HelpLine$ = "Edit keys, allowed: Home, Left, Right, End, Backspace, Del, Esc"
03,758 LOCATE TopLrow + 41 - 23 - shrunkP, TopLcol + 2: PRINT HelpLine$
03,759 _DISPLAY
03,760 DrawBoxShadow3exact$ = InputLine$(TopLrow + 39 - 23 - shrunkP, TopLcol + 2, BottomRcol - TopLcol - 1 - 2)
03,761 LOCATE 1, 1, 1, 30, 31
03,762
03,763 END FUNCTION
03,764
03,765 FUNCTION DrawBoxShadow3exactRussian$ (TopLrow, TopLcol, BottomRow, BottomRcol, Captmn$)
03,766     SHARED FileSize
03,767     SHARED LoadedFile$
03,768     SHARED PSPlike$
03,769     SHARED YdimROW, filecount, File_Frame_y, XdimCOL, FileArrayWINDOW$( ), FileArray$( )
03,770     'Shadow
03,771     BckGRcolor = 5
03,772     COLOR 8, BACKGR
03,773     FOR i = 1 TO MIN$(YdimROW, filecount)
03,774         'DumboReadOnceNotThrice$ = FileArray$(i + (File_Frame_y - 1))
03,775         'IF LEN(DumboReadOnceNotThrice$) >= XdimCOL THEN
03,776             'FileArrayWINDOW$(i) = MID$(DumboReadOnceNotThrice$, 1, XdimCOL)
03,777         'ELSE
03,778             'FileArrayWINDOW$(i) = DumboReadOnceNotThrice$ + SPACE$(XdimCOL - LEN(DumboReadOnceNotThrice$))
03,779         'END IF
03,780         IF i >= TopLrow + 1 AND i <= BottomRow + 1 THEN
03,781             LOCATE i, TopLcol + 1: PRINT MID$(FileArrayWINDOW$(i), TopLcol + 1, BottomRcol - TopLcol + 1);
03,782         END IF
03,783     NEXT
03,784     'Outer frame
03,785     LOCATE TopLrow, TopLcol
03,786     COLOR 7, BckGRcolor
03,787     PRINT CHR$(218); STRING$(BottomRcol - TopLcol - 1, CHR$(196));
03,788     COLOR 0, BckGRcolor: PRINT CHR$(191);
03,789     COLOR 7, BckGRcolor: LOCATE TopLrow, TopLcol + 2: PRINT "[ ";: COLOR 15, BckGRcolor: PRINT Captmn$;: COLOR 7, BckGRcolor: PRINT " ]";
03,790     FOR i = TopLrow + 1 TO BottomRow - 1
03,791         COLOR 7, BckGRcolor: LOCATE i, TopLcol: PRINT CHR$(179); STRING$(BottomRcol - TopLcol - 1, CHR$(32));: COLOR 0, BckGRcolor: PRINT CHR$(179);
03,792     NEXT
03,793     LOCATE BottomRow, TopLcol
03,794     COLOR 7, BckGRcolor: PRINT CHR$(192);
03,795     COLOR 0, BckGRcolor: PRINT STRING$(BottomRcol - TopLcol - 1, CHR$(196)); CHR$(217)
03,796     'Inner frame
03,797     'In case of YdimROW = 40 then shrunk panel
03,798     shrunkP = 14
03,799     LOCATE TopLrow + 38 - 23 - shrunkP, TopLcol + 1
03,800     COLOR 0, BckGRcolor
03,801     PRINT CHR$(218); STRING$(BottomRcol - 1 - (TopLcol + 1) - 1, CHR$(196));
03,802     COLOR 7, BckGRcolor: PRINT CHR$(191);
03,803     FOR i = TopLrow + 39 - 23 - shrunkP TO TopLrow + 40 - 23 - shrunkP
03,804         COLOR 0, BckGRcolor: LOCATE i, TopLcol + 1: PRINT CHR$(179); STRING$(BottomRcol - 1 - (TopLcol + 1) - 1, CHR$(32));: COLOR 7, BckGRcolor: PRINT CHR$(179);
03,805     NEXT

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

03,806 LOCATE TopLrow + 40 - 23 - shrunkP, TopLcol + 1
03,807 COLOR 0, BckGRcolor: PRINT CHR$(192);
03,808 COLOR 7, BckGRcolor: PRINT STRING$(BottomRcol - 1 - (TopLcol + 1) - 1, CHR$(196)); CHR$(217)
03,809
03,810 LOCATE TopLrow + 2, TopLcol + 2
03,811 COLOR 7, BckGRcolor
03,812
03,813 HelpLine$ = "Edit keys, allowed: Home, Left, Right, End, Backspace, Del, Esc"
03,814 LOCATE TopLrow + 41 - 23 - shrunkP, TopLcol + 2: PRINT HelpLine$
03,815 _DISPLAY
03,816 DrawBoxShadow3exactRussian$ = InputLineRussian$(TopLrow + 39 - 23 - shrunkP, TopLcol + 2, BottomRcol - TopLcol - 1 - 2)
03,817 LOCATE 1, 1, 1, 30, 31
03,818
03,819 END FUNCTION
03,820
03,821 FUNCTION InputLine$ (Irow, Icol, LineLen)
03,822 STATIC a$
03,823 SHARED CurrentWord$
03,824 LOCATE Irow, Icol, 1, 30, 31
03,825 MaxCol = Icol + LineLen - 1
03,826 CurCol = Icol 'Icol..MaxCol
03,827 'IF CurrentWord$ = "" THEN
03,828 ' IF INSTR(_CLIPBOARD$, CHR$(10)) THEN ' _CLIPBOARD$ could be multi-line, so get the first
03,829 ' a2$ = LEFT$(_CLIPBOARD$, INSTR(_CLIPBOARD$, CHR$(10)) - 1)
03,830 ' IF RIGHT$(a2$, 1) = CHR$(13) THEN a2$ = LEFT$(a2$, LEN(a2$) - 1)
03,831 ' a$ = LEFT$(a2$, LineLen - 1) '""
03,832 ' ELSE
03,833 ' a$ = LEFT$(_CLIPBOARD$, LineLen - 1) '""
03,834 ' END IF
03,835 ' CurCol = Icol + LEN(a$)
03,836 'ELSE
03,837 ' a$ = CurrentWord$
03,838 'END IF
03,839 DO
03,840 key$ = INKEY$
03,841 IF key$ <> "" THEN
03,842 code% = ASC(key$):
03,843 IF code% THEN ' ASC returns any value greater than 0
03,844 SELECT CASE ASC(key$)
03,845 CASE 27:
03,846 a$ = "": CurCol = Icol
03,847 CASE 13:
03,848 IF a$ <> "" THEN EXIT DO
03,849 CASE 8: 'Backspace
03,850 IF CurCol <> Icol THEN
03,851 IF CurCol = Icol + LEN(a$) THEN ' this condition includes a$ = ""
03,852 a$ = LEFT$(a$, LEN(a$) - 1)
03,853 ELSE 'inhere we are not at the end i.e. 1..len(a$)-1
03,854 a$ = LEFT$(a$, (CurCol - Icol) - 1) + RIGHT$(a$, LEN(a$) - (CurCol - Icol))

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

03,855         END IF
03,856         CurCol = CurCol - 1
03,857     END IF
03,858     CASE 32 TO 127 - 1:
03,859         IF LEN(a$) < LineLen - 1 THEN 'a$ = a$ + key$: CurCol = CurCol + 1 '-1 because cursor goes next to the field otherwise
03,860         IF CurCol = Icol + LEN(a$) THEN ' this condition includes a$ = ""
03,861             a$ = a$ + key$
03,862         ELSE 'inhere we are not at the end i.e. 1..len(a$)-1
03,863             ' Insert at (CurCol - Icol) + 1:
03,864             a$ = LEFT$(a$, (CurCol - Icol)) + key$ + RIGHT$(a$, LEN(a$) - (CurCol - Icol))
03,865         END IF
03,866         CurCol = CurCol + 1
03,867     END IF
03,868 END SELECT
03,869 ELSE
03,870     SELECT CASE ASC(key$, 2)
03,871         CASE 75: IF CurCol > Icol THEN CurCol = CurCol - 1
03,872         CASE 77: IF CurCol < MaxCol AND CurCol < Icol + LEN(a$) THEN CurCol = CurCol + 1
03,873         CASE 71: CurCol = Icol 'Home
03,874         CASE 79: CurCol = Icol + LEN(a$) 'End
03,875         CASE 83: 'Del
03,876             IF a$ <> "" THEN
03,877                 a$ = LEFT$(a$, (CurCol - Icol)) + RIGHT$(a$, LEN(a$) - (CurCol - Icol) - 1)
03,878             END IF
03,879         CASE 82: 'Ins
03,880     END SELECT
03,881 END IF
03,882 END IF
03,883 LOCATE Irow, Icol, 1, 30, 31
03,884 PRINT STRING$(LineLen, " ");
03,885 LOCATE Irow, Icol, 1, 30, 31
03,886 PRINT a$;
03,887 LOCATE Irow, CurCol, 1, 30, 31
03,888 _DISPLAY
03,889 _LIMIT 30
03,890 LOOP
03,891 'DO WHILE _KEYDOWN(13): LOCATE , , 0: _DISPLAY: LOOP
03,892 _KEYCLEAR
03,893 InputLine$ = a$
03,894 END FUNCTION
03,895
03,896 FUNCTION InputLineRussian$ (Irow, Icol, LineLen)
03,897     STATIC a$
03,898     SHARED CurrentWord$
03,899     LOCATE Irow, Icol, 1, 30, 31
03,900     MaxCol = Icol + LineLen - 1
03,901     CurCol = Icol 'Icol..MaxCol
03,902     'IF CurrentWord$ = "" THEN
03,903         ' IF INSTR(_CLIPBOARD$, CHR$(10)) THEN ' _CLIPBOARD$ could be multi-line, so get the first

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,904 '      a2$ = LEFT$(_CLIPBOARD$, INSTR(_CLIPBOARD$, CHR$(10)) - 1)
03,905 '      IF RIGHT$(a2$, 1) = CHR$(13) THEN a2$ = LEFT$(a2$, LEN(a2$) - 1)
03,906 '      a$ = LEFT$(a2$, LineLen - 1) '""
03,907 '      ELSE
03,908 '      a$ = LEFT$(_CLIPBOARD$, LineLen - 1) '""
03,909 '      END IF
03,910 '      CurCol = Icol + LEN(a$)
03,911 'ELSE
03,912 '      a$ = CurrentWord$
03,913 'END IF
03,914 DO
03,915     key$ = INKEY$
03,916     IF key$ <> "" THEN
03,917         code% = ASC(key$):
03,918         IF code% THEN ' ASC returns any value greater than 0
03,919             SELECT CASE ASC(key$)
03,920                 CASE 27:
03,921                     a$ = "": CurCol = Icol
03,922                 CASE 13:
03,923                     IF a$ <> "" THEN EXIT DO
03,924                 CASE 8: 'Backspace
03,925                     IF CurCol <> Icol THEN
03,926                         IF CurCol = Icol + LEN(a$) THEN ' this condition includes a$ = ""
03,927                             a$ = LEFT$(a$, LEN(a$) - 1)
03,928                         ELSE 'inhere we are not at the end i.e. 1..len(a$)-1
03,929                             a$ = LEFT$(a$, (CurCol - Icol) - 1) + RIGHT$(a$, LEN(a$) - (CurCol - Icol))
03,930                         END IF
03,931                         CurCol = CurCol - 1
03,932                     END IF
03,933                 CASE 32 TO 127 - 1:
03,934                     k$ = key$
03,935                     IF k$ = "~" THEN k$ = CHR$(250)
03,936                     IF k$ = "`" THEN k$ = CHR$(251)
03,937
03,938                     IF k$ = "+" THEN k$ = CHR$(154)
03,939                     IF k$ = "=" THEN k$ = CHR$(234)
03,940
03,941                     IF k$ = "Q" THEN k$ = CHR$(159)
03,942                     IF k$ = "q" THEN k$ = CHR$(239)
03,943
03,944                     IF k$ = "W" THEN k$ = CHR$(152)
03,945                     IF k$ = "w" THEN k$ = CHR$(232)
03,946
03,947                     IF k$ = "E" THEN k$ = CHR$(133)
03,948                     IF k$ = "e" THEN k$ = CHR$(165)
03,949
03,950                     IF k$ = "R" THEN k$ = CHR$(144)
03,951                     IF k$ = "r" THEN k$ = CHR$(224)
03,952

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```

03,953      IF k$ = "T" THEN k$ = CHR$(146)
03,954      IF k$ = "t" THEN k$ = CHR$(226)
03,955
03,956      IF k$ = "Y" THEN k$ = CHR$(155)
03,957      IF k$ = "y" THEN k$ = CHR$(235)
03,958
03,959      IF k$ = "U" THEN k$ = CHR$(147)
03,960      IF k$ = "u" THEN k$ = CHR$(227)
03,961
03,962      IF k$ = "I" THEN k$ = CHR$(136)
03,963      IF k$ = "i" THEN k$ = CHR$(168)
03,964
03,965      IF k$ = "O" THEN k$ = CHR$(142)
03,966      IF k$ = "o" THEN k$ = CHR$(174)
03,967
03,968      IF k$ = "P" THEN k$ = CHR$(143)
03,969      IF k$ = "p" THEN k$ = CHR$(175)
03,970
03,971      IF k$ = "{" THEN k$ = CHR$(158)
03,972      IF k$ = "[" THEN k$ = CHR$(238)
03,973
03,974      IF k$ = "}" THEN k$ = CHR$(153)
03,975      IF k$ = "]" THEN k$ = CHR$(233)
03,976
03,977      IF k$ = "!" THEN k$ = CHR$(157)
03,978      IF k$ = "\" THEN k$ = CHR$(237)
03,979
03,980      IF k$ = "A" THEN k$ = CHR$(128)
03,981      IF k$ = "a" THEN k$ = CHR$(160)
03,982
03,983      IF k$ = "S" THEN k$ = CHR$(145)
03,984      IF k$ = "s" THEN k$ = CHR$(225)
03,985
03,986      IF k$ = "D" THEN k$ = CHR$(132)
03,987      IF k$ = "d" THEN k$ = CHR$(164)
03,988
03,989      IF k$ = "F" THEN k$ = CHR$(148)
03,990      IF k$ = "f" THEN k$ = CHR$(228)
03,991
03,992      IF k$ = "G" THEN k$ = CHR$(131)
03,993      IF k$ = "g" THEN k$ = CHR$(163)
03,994
03,995      IF k$ = "H" THEN k$ = CHR$(151)
03,996      IF k$ = "h" THEN k$ = CHR$(231)
03,997
03,998      IF k$ = "J" THEN k$ = CHR$(137)
03,999      IF k$ = "j" THEN k$ = CHR$(169)
04,000
04,001      IF k$ = "K" THEN k$ = CHR$(138)

```

```

04,002         IF k$ = "k" THEN k$ = CHR$(170)
04,003
04,004         IF k$ = "l" THEN k$ = CHR$(139)
04,005         IF k$ = "1" THEN k$ = CHR$(171)
04,006
04,007         IF k$ = ":" THEN k$ = CHR$(156)
04,008         IF k$ = ";" THEN k$ = CHR$(236)
04,009
04,010         IF k$ = CHR$(34) THEN k$ = CHR$(134)
04,011         IF k$ = "'" THEN k$ = CHR$(166)
04,012
04,013         IF k$ = "Z" THEN k$ = CHR$(135)
04,014         IF k$ = "z" THEN k$ = CHR$(167)
04,015
04,016         IF k$ = "X" THEN k$ = CHR$(149)
04,017         IF k$ = "x" THEN k$ = CHR$(229)
04,018
04,019         IF k$ = "C" THEN k$ = CHR$(150)
04,020         IF k$ = "c" THEN k$ = CHR$(230)
04,021
04,022         IF k$ = "V" THEN k$ = CHR$(130)
04,023         IF k$ = "v" THEN k$ = CHR$(162)
04,024
04,025         IF k$ = "B" THEN k$ = CHR$(129)
04,026         IF k$ = "b" THEN k$ = CHR$(161)
04,027
04,028         IF k$ = "N" THEN k$ = CHR$(141)
04,029         IF k$ = "n" THEN k$ = CHR$(173)
04,030
04,031         IF k$ = "M" THEN k$ = CHR$(140)
04,032         IF k$ = "m" THEN k$ = CHR$(172)
04,033
04,034         key$ = k$
04,035         IF LEN(a$) < LineLen - 1 THEN 'a$ = a$ + key$: CurCol = CurCol + 1 '-1 because cursor goes next to the field otherwise
04,036             IF CurCol = Icol + LEN(a$) THEN ' this condition includes a$ = ""
04,037                 a$ = a$ + key$
04,038             ELSE 'inhere we are not at the end i.e. 1..len(a$)-1
04,039                 ' Insert at (CurCol - Icol) + 1:
04,040                 a$ = LEFT$(a$, (CurCol - Icol)) + key$ + RIGHT$(a$, LEN(a$) - (CurCol - Icol))
04,041             END IF
04,042             CurCol = CurCol + 1
04,043         END IF
04,044     END SELECT
04,045 ELSE
04,046     SELECT CASE ASC(key$, 2)
04,047         CASE 75: IF CurCol > Icol THEN CurCol = CurCol - 1
04,048         CASE 77: IF CurCol < MaxCol AND CurCol < Icol + LEN(a$) THEN CurCol = CurCol + 1
04,049         CASE 71: CurCol = Icol 'Home
04,050         CASE 79: CurCol = Icol + LEN(a$) 'End

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

04,051         CASE 83: 'Del
04,052             IF a$ <> "" THEN
04,053                 a$ = LEFT$(a$, (CurCol - Icol)) + RIGHT$(a$, LEN(a$) - (CurCol - Icol) - 1)
04,054             END IF
04,055         CASE 82: 'Ins
04,056     END SELECT
04,057 END IF
04,058 END IF
04,059 LOCATE Irow, Icol, 1, 30, 31
04,060 PRINT STRING$(LineLen, " ");
04,061 LOCATE Irow, Icol, 1, 30, 31
04,062 PRINT a$;
04,063 LOCATE Irow, CurCol, 1, 30, 31
04,064 _DISPLAY
04,065 _LIMIT 30
04,066 LOOP
04,067 'DO WHILE _KEYDOWN(13): LOCATE , , 0: _DISPLAY: LOOP
04,068 _KEYCLEAR
04,069 InputLineRussian$ = a$
04,070 END FUNCTION
04,071
04,072 SUB NextFrame (TopRow)
04,073     SHARED ASCIIIFrame
04,074     SHARED PSPlike$
04,075     LOCATE TopRow, 73
04,076     ASCIIIFrame = ASCIIIFrame + 1
04,077     IF ASCIIIFrame > 20 THEN ASCIIIFrame = 1 '1..20
04,078     f = FREEFILE
04,079     fr$ = LTRIM$(STR$(ASCIIIFrame))
04,080     IF LEN(fr$) = 1 THEN fr$ = "0" + fr$
04,081     IF _FILEEXISTS(PSPlike$ + "glass" + fr$ + ".txt") THEN
04,082         OPEN PSPlike$ + "glass" + fr$ + ".txt" FOR INPUT AS #f
04,083         DO WHILE NOT EOF(f)
04,084             LINE INPUT #f, a$
04,085             LOCATE , 73
04,086             PRINT a$
04,087         LOOP
04,088         CLOSE #f
04,089     END IF
04,090 END SUB
04,091
04,092 SUB NextFramePEN (TopRow)
04,093     SHARED ASCIIIFramePEN
04,094     SHARED PSPlike$
04,095     SHARED XdimCOL
04,096     IF XdimCOL = 198 THEN HorizPos = 128 ELSE HorizPos = 70
04,097     LOCATE TopRow, HorizPos
04,098     ASCIIIFramePEN = ASCIIIFramePEN + 1
04,099     IF ASCIIIFramePEN > 24 THEN ASCIIIFramePEN = 1 '1..20

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

04,100 f = FREEFILE
04,101 fr$ = LTRIM$(STR$(ASCIIFramePEN))
04,102 IF LEN(fr$) = 1 THEN fr$ = "0" + fr$
04,103 IF _FILEEXISTS(PSPlike$ + "pen" + fr$ + ".txt") THEN
04,104     OPEN PSPlike$ + "pen" + fr$ + ".txt" FOR INPUT AS #f
04,105     DO WHILE NOT EOF(f)
04,106         LINE INPUT #f, a$
04,107         LOCATE , HorizPos
04,108         PRINT RIGHT$(a$, 80 - 23)
04,109     LOOP
04,110     CLOSE #f
04,111 END IF
04,112 END SUB
04,113
04,114 SUB ShowF1
04,115     PostFix$ = ""
04,116     'IF ToLoadOrNotFlag = 0 THEN PostFix$ = PostFix$ + "_External" ELSE PostFix$ = PostFix$ + "_Fast"
04,117     IF WrapFlag = 1 THEN PostFix$ = PostFix$ + "_Wrapper" ELSE PostFix$ = PostFix$ + "_Vanilla"
04,118     PRINT "Masakari, revision 8.1+" + PostFix$ + ", written in QB64 by Kaze, source code downloadable at https://www.qb64.org/forum"
04,119
04,120     $IF WINDOWS THEN
04,121         PRINT "Usage: Masakari [filename]![/help][-ascii[_gesch]]"
04,122     $ELSE
04,123         Print "Usage: Masakari [filename]![-h][-ascii[_gesch]]"
04,124     $END IF
04,125     PRINT "Note: The 'filename' could be a filelist, i.e. a list of filenames (see Space and Double-Left-Click)."
```

04,126 PRINT

04,127 PRINT "Mouse:"

04,128 PRINT " Button 1 - sets the cursor and the inverse line to the chosen position; hold (without moving for 4 seconds) to quit"

04,129 PRINT " Button 2 - [when holded] Draws"

04,130 PRINT " Button 3 - Burst PgDn"

04,131 PRINT " Wheel Up/Dn - Up/Dn"

04,132 PRINT " Button 2 + Wheel Up/Dn - going to the top/bottom left position"

04,133 PRINT " Button 1 + Button 2 - creating PAGODA file for the Current_Word, the loaded file must be in Current Directory!"

04,134 PRINT " Button 2 + dragging (sideways) for at least 100 columns/cells (within 2 seconds) - same as Alt+X, Alt+Q"

04,135 PRINT " Button 2 + dragging (from top to bottom) for at least 5 lines/cells (within 2 seconds) - same as PgUp"

04,136 PRINT " Button 2 + dragging (from bottom to top) for at least 5 lines/cells (within 2 seconds) - same as PgDn"

04,137 PRINT " Button 1 + Button 1 (within 0.33 seconds) - Double-Left-Click - PgUp or loads the inverse line (if it is a valid file)"

04,138 PRINT " Button 2 + Button 2 (within 0.33 seconds) - Double-Right-Click - PgDn"

04,139 PRINT " LShift + Button 1/2 - Burst Up/Dn"

04,140 PRINT " LCtrl + Button 1/2 - Burst PgUp/PgDn"

04,141 PRINT " LAlt + Button 1/2 = LCtrl+Home - going to the top left position / LCtrl+End - going to the bottom left position"

04,142 PRINT "Keyboard:"

04,143 PRINT " F1 - this help panel; Enter (released) - opens the current/INVERSE line, WRAPPED, into a new scroll window"

04,144 PRINT " Up/Dn/PgUp/PgDn"

04,145 PRINT " Left/Right - still no sideways scroll"

04,146 PRINT " LCtrl+Home/End - going to the top/bottom left position"

04,147 PRINT " Alt+X or Alt+Q - quit to the system, without demanding keypress"

04,148 PRINT " Space - loads the highlighted line (if it is an actual filename)"

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

04,149 PRINT "   BCtrl+RShift - (can hold it) highlighting all unfamiliar words to masakari.wrd file/wordlist"
04,150 PRINT "   LCtrl+F3 - Search Panel, Kazahana (with 16 threads) searches Externally the file, currently displayed"
04,151 PRINT "   LAlt+F3 released - Non-Interactive search for Current_Word, from the line below the INVERSE one, downward"
04,152 PRINT "   LShift+F3 released - Non-Interactive search for CLIPBOARD, from the line below the INVERSE one, downward"
04,153 PRINT "   F2/F3 released - Search Panel, Interactive search for User_Input, from the line below the INVERSE one, downward"
04,154 'Note: The [LShift+]F3 search modes are Exact and Case-Sensitive, non-whole word as well
04,155 PRINT "Benchmarking:"
04,156 PRINT "       LAlt+BAlt - Reporting (in the status line in red color) the time for load"
04,157 PRINT "       LCtrl+RCtrl - Reporting (in the status line in red color) the time for PgDn-ing (page-by-page) the entire file"
04,158 PRINT "       LShift+RShift - Reporting (in the status line in red color) the time for Down-ing (line-by-line) the entire file"
04,159 PRINT "       Esc - Stops the auto-scrolling started with LCtrl+RCtrl or LShift+RShift"
04,160 PRINT "Note1: During the LShift+RShift auto-scroll, you can still press/hold BCtrl, during the LCtrl+RCtrl, you cannot."
04,161 PRINT "Note2: During the auto-scroll, you can still press PgUp, PgDn, LCtrl+Home, LCtrl+End, Esc."
04,162 PRINT "Note3: The 'Vanilla' sub-variant loads textual files without wrapping the lines."
04,163 PRINT "Note4: The 'Wrapper' sub-variant makes the text file viewable without side/lateral scroll."
04,164 PRINT "       If unwrappable lines exist then those lines are dumped to filename+"; CHR$(34); ".unwrappable"; CHR$(34); ", "
04,165 PRINT "       otherwise, the wrapped lines are dumped to filename+"; CHR$(34); ".wrapped"; CHR$(34); ", and auto-loaded."
04,166 PRINT "       If wrapped file exists during start then it is used, not re-created."
04,167 END SUB
04,168
04,169 FUNCTION timeElapsedSince! (startTime!)
04,170     IF startTime! > TIMER THEN startTime! = startTime! - 86400
04,171     timeElapsedSince! = TIMER - startTime!
04,172 END FUNCTION
04,173
04,174 FUNCTION CROPorPADatRIGHT$ (victim$, linelen)
04,175     IF LEN(victim$) > linelen THEN
04,176         CROPorPADatRIGHT$ = LEFT$(victim$, linelen)
04,177     ELSE
04,178         CROPorPADatRIGHT$ = victim$ + SPACE$(linelen - LEN(victim$))
04,179     END IF
04,180 END FUNCTION
04,181
04,182 'CONST LSHIFTkey& = 100304
04,183 'DO
04,184 '   IF _KEYDOWN(LSHIFTkey&) = 0 AND IsF3released THEN PRINT "F3 released ";
04,185 '   _LIMIT 500
04,186 'LOOP
04,187 FUNCTION IsF3released
04,188     IsF3released = 0
04,189     MustBeLONGssigned& = _KEYHIT
04,190     IF MustBeLONGssigned& THEN
04,191         IF MustBeLONGssigned& < 0 THEN 'negative value means key released
04,192             MustBeLONGssigned& = -MustBeLONGssigned&
04,193             IF MustBeLONGssigned& \ 256 = 61 THEN
04,194                 IsF3released = 1
04,195             ELSE
04,196                 END IF
04,197     END IF

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxL2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

04,296 '      FOR MumboJumbo = 1 TO 4
04,297 '      PRINT "NEW"; Button1LOG_firstDetection#(MumboJumbo), Button1LOG_ForHowLongHolded#(MumboJumbo)
04,298 '      NEXT
04,299
04,300 '  END IF
04,301
04,302 '  IF Button1LOG_firstDetection#(3) AND (Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(3) < DclickTime#) THEN Double_LeftClick = 1 ELSE Double_LeftClick = 0
04,303 '  IF Double_LeftClick = 1 THEN PRINT "Double_LeftClick detected. Done in"; INT((Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(3)) * 1000); "ms."
04,304
04,305 '  IF Button1LOG_firstDetection#(2) AND (Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(2) < DclickTime# * 2) THEN Triple_LeftClick = 1 ELSE Triple_LeftClick = 0
04,306 '  IF Triple_LeftClick = 1 THEN PRINT "Triple_LeftClick detected. Done in"; INT((Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(2)) * 1000); "ms."
04,307
04,308 '  IF Button1LOG_firstDetection#(1) AND (Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(1) < DclickTime# * 3) THEN Quadruple_LeftClick = 1 ELSE Quadruple_LeftClick = 0
04,309 '  IF Quadruple_LeftClick = 1 THEN PRINT "Quadruple_LeftClick detected. Done in"; INT((Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(1)) * 1000); "ms."
04,310
04,311 '  IF (ForHowLongWasPressed1# > ShutDownDuration#) THEN
04,312 '    ShutDown_LeftClick = 1: IF ShutDown_LeftClick = 1 THEN PRINT "ShutDown_LeftClick detected."; END
04,313 '  END IF
04,314
04,315 '  _LIMIT 500
04,316 '  _DISPLAY
04,317 'LOOP
04,318
04,319 'Note1: From above experiments I see Double-Click fits usually in DclickTime# s window.
04,320 'Note2: Double-Click is detected when the difference between two initial clicks is below DclickTime# s i.e. Button1LOG_firstDetection#(4)-Button1LOG_firstDetection#(3) < DclickTime#
04,321 ' ForHowLongWasPressed1# = B1 - A1 i.e. duration of holding down:
04,322 ' where
04,323 ' A1=first detection of the click  B1=released A2=first detection of the click  B2=released
04,324 ' [           ][           ][           ][           ]
04,325
04,326 'mouse mumbo-jumbo ]
04,327
04,328 ' The need for more ergonomic (that is, easily accessible) shortcuts (strictly keyboardish) got me thinking...
04,329 ' Holy hell, why no one introduced the analogues/counterparts of "double-clicks" - the first that comes to mind: the "double-hits".
04,330 ' Both, being just taps.
04,331 ' Currently, in Masakari there are:
04,332 ' - shortcuts, purely keyboardish;
04,333 ' - shortcuts, mix of keys and mouse buttons;
04,334 ' - shortcuts, purely mouseish, just mouse stuff - buttons and wheel.
04,335 ' From revision 8.1+ onward, following double-hits were implemented:
04,336 ' Double LShift; Double LCtrl; Double LAlt; Double RAlt.
04,337 ' The thing is that "legacy" shortcuts as Ctrl+F3 (even the F3) force the user to "locate" their position and to overreach - meaning hovering pass the SPACEBAR area - which is most easy to hit since the wrists/palms have support, and are simply closer.
04,338 ' To me, the traditional key sequences are to be replaced/enriched with more convenient, more close to the first row of the keyboard, ones.
04,339 ' Currently, my main keyboard is a nasty one (laptopish, with low profile, non-tactile, half-sized F-keys, and tightly packed), when e.g. a LCtrl+F3 search is needed, spotting this halved F3 is a drag, therefore the double-hit of e.g. LCtrl saves the situation - just double-tap it - the timings between the two pressings are handled by the same logic as the mouse double/triple/quadruple-clicks.
04,340 ' The following stand-alone etude is the same as the previously shared mouse_mumbo-jumbo.bas, except:
04,341 ' buttowndown1 = _MOUSEBUTTON(1)
04,342 ' should be replaced with

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

04,343 ' KeyTap_buttondown1 = _KEYDOWN(LSHIFTkey&)
04,344
04,345 ''keyboard mumbo-jumbo [
04,346
04,347 'CONST RSHIFTkey& = 100303
04,348 'CONST LSHIFTkey& = 100304
04,349 'CONST RCTRLkey& = 100305
04,350 'CONST LCTRLkey& = 100306
04,351 'CONST RALTkey& = 100307
04,352 'CONST LALTkey& = 100308
04,353
04,354 'CONST BACKSPCkey& = 8
04,355 'CONST TABkey& = 9
04,356 'CONST SPACEkey& = 32
04,357 'CONST ESCkey& = 27
04,358 'CONST ENTERkey& = 13
04,359
04,360 'CONST HOMEkey& = 18176
04,361 'CONST ENDkey& = 20224
04,362
04,363 'CONST INSkey& = 20992
04,364 'CONST DELkey& = 21248
04,365
04,366 'CONST PGUPkey& = 18688
04,367 'CONST PGDNkey& = 20736
04,368
04,369 'CONST LEFTkey& = 19200
04,370 'CONST RIGHTkey& = 19712
04,371 'CONST UPkey& = 18432
04,372 'CONST DOWNkey& = 20480
04,373
04,374 'SCREEN _NEWIMAGE(600, 900, 256)
04,375 '_SCREENMOVE 32, 32
04,376
04,377 'ShutDownDuration# = 4 ' 4 seconds
04,378 'DclickTime# = 0.33 ' 1/3 of a second, usually it is 0.27s, so 0.33 suits even the slow clickers
04,379 'DIM Button1LOG_firstDetection#(4)
04,380 'DIM Button1LOG_ForHowLongHoled#(4)
04,381 'PrevClick1# = 0
04,382 'PRINT "You may exit with Esc or by holding left Shift key for 4 seconds..."
04,383 'DO 'WHILE INKEY$ <> CHR$(27)
04,384
04,385 ' 'AsIfItIsINKEY% = _MOUSEINPUT ' Check the mouse status
04,386 ' IF AsIfItIsINKEY% THEN
04,387 ' 'buttondown1 = _MOUSEBUTTON(1)
04,388 ' 'buttondown2 = _MOUSEBUTTON(2)
04,389 ' 'buttondown3 = _MOUSEBUTTON(3)
04,390 ' 'mwheel = _MOUSEWHEEL
04,391 ' END IF

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

04,392
04,393 '   buttondown1 = _KEYDOWN(LSHIFTkey&)
04,394 '   buttondown2 = _KEYDOWN(LCTRLkey&)
04,395 '   buttondown3 = _KEYDOWN(LALTkey&)
04,396
04,397 '   IF (buttondown1 AND PrevClick1# = 0) THEN 'first detection of the click
04,398 '       PrevClick1# = TIMER(0.001)
04,399 '   END IF
04,400
04,401 '   IF (buttondown1 AND PrevClick1# <> 0) THEN 'already clicked
04,402 '       PrevClick1# = TIMER(0.001)
04,403 '       ForHowLongWasPressed1# = TIMER(0.001) - PrevClick1#
04,404 '   END IF
04,405
04,406 '   IF buttondown1 = 0 AND PrevClick1# <> 0 THEN 'write to the log
04,407 '       PRINT
04,408 '       FOR MumboJumbo = 1 TO 4
04,409 '           PRINT "OLD"; Button1LOG_firstDetection#(MumboJumbo), Button1LOG_ForHowLongHoded#(MumboJumbo)
04,410 '       NEXT
04,411
04,412 '       FOR MumboJumbo = 1 TO 3
04,413 '           Button1LOG_firstDetection#(MumboJumbo) = Button1LOG_firstDetection#(MumboJumbo + 1)
04,414 '           Button1LOG_ForHowLongHoded#(MumboJumbo) = Button1LOG_ForHowLongHoded#(MumboJumbo + 1)
04,415 '       NEXT
04,416 '       Button1LOG_firstDetection#(4) = PrevClick1#
04,417 '       Button1LOG_ForHowLongHoded#(4) = ForHowLongWasPressed1#
04,418 '       PrevClick1# = 0
04,419
04,420 '       FOR MumboJumbo = 1 TO 4
04,421 '           PRINT "NEW"; Button1LOG_firstDetection#(MumboJumbo), Button1LOG_ForHowLongHoded#(MumboJumbo)
04,422 '       NEXT
04,423
04,424 '   END IF
04,425
04,426 '   IF Button1LOG_firstDetection#(3) AND (Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(3) < DclickTime#) THEN Double_LeftClick = 1 ELSE Double_LeftClick = 0
04,427 '   IF Double_LeftClick = 1 THEN PRINT "Double_LeftShift detected. Done in"; INT((Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(3)) * 1000); "ms."
04,428
04,429 '   IF Button1LOG_firstDetection#(2) AND (Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(2) < DclickTime# * 2) THEN Triple_LeftClick = 1 ELSE Triple_LeftClick = 0
04,430 '   IF Triple_LeftClick = 1 THEN PRINT "Triple_LeftShift detected. Done in"; INT((Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(2)) * 1000); "ms."
04,431
04,432 '   IF Button1LOG_firstDetection#(1) AND (Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(1) < DclickTime# * 3) THEN Quadruple_LeftClick = 1 ELSE Quadruple_LeftClick = 0
04,433 '   IF Quadruple_LeftClick = 1 THEN PRINT "Quadruple_LeftShift detected. Done in"; INT((Button1LOG_firstDetection#(4) - Button1LOG_firstDetection#(1)) * 1000); "ms."
04,434
04,435 '   IF (ForHowLongWasPressed1# > ShutDownDuration#) THEN
04,436 '       ShutDown_LeftClick = 1: IF ShutDown_LeftClick = 1 THEN PRINT "ShutDown_LeftShift detected.": END
04,437 '   END IF
04,438
04,439 '   _LIMIT 500
04,440 '   _DISPLAY

```

```
04,441 'LOOP
04,442
04,443 ''Note1: From above experiments I see Double-Click fits usually in DclickTime# s window.
04,444 ''Note2: Double-Click is detected when the difference between two initial clicks is below DclickTime# s i.e. Button1LOG_firstDetection#(4)-Button1LOG_firstDetection#(3) < DclickTime#
04,445 '' ForHowLongWasPressed1# = B1 - A1 i.e. duration of holding down:
04,446 '' where
04,447 '' A1=first detection of the click   B1=released A2=first detection of the click   B2=released
04,448 '' [                               ][                               ][                               ]
04,449
04,450 ''keyboard mumbo-jumbo ]
04,451
```

UTF8toGesch.bas:

00,001 REM Converting UTF8 to Gesch codepage

00,002

00,003 \$SCREENHIDE

00,004 \$CONSOLE

00,005 _CONSOLE ON

00,006 _CONSOLETITLE "Gesch codepage convertor, revision 2"

00,007 _DEST _CONSOLE

00,008

00,009 REM Microsoft_windows_cp1251: 'Cyrillic alphabet such as Russian, Bulgarian, Serbian Cyrillic and other languages. It is the most widely used for encoding the Bulgarian, Serbian and Macedonian languages.

00,010 REM 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143

00,011 REM DATA 1026,1027,8218,1107,8222,8230,8224,8225,8364,8240,1033,8249,1034,1036,1035,1039 '128+16*0 to 128+16*1-1

00,012 REM 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159

00,013 REM DATA 1106,8216,8217,8220,8221,8226,8211,8212,0 ,8482,1113,8250,1114,1116,1115,1119 '128+16*1 to 128+16*2-1

00,014 REM 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175

00,015 REM DATA 160 ,1038,1118,1032,164 ,1168,166 ,167 ,1025,169 ,1028,171 ,172 ,173 ,174 ,1031 '128+16*2 to 128+16*3-1

00,016 REM 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191

00,017 REM DATA 176 ,177 ,1030,1110,1169,181 ,182 ,183 ,1105,8470,1108,187 ,1112,1029,1109,1111 '128+16*3 to 128+16*4-1

00,018 REM Cyrillic:

00,019 REM DATA 1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055 '128+16*4 to 128+16*5-1

00,020 REM DATA 1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071 '128+16*5 to 128+16*6-1

00,021 REM DATA 1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087 '128+16*6 to 128+16*7-1

00,022 REM DATA 1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103 '128+16*7 to 128+16*8-1 --\

00,023 REM

00,024 REM Microsoft_pc_cpGESCH: 'Gesch is Sanmayce's layout, combining the MIK and 437, in this way:

00,025 REM

00,026 REM DATA 1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055 '128+16*0 to 128+16*1-1

00,027 REM DATA 1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071 '128+16*1 to 128+16*2-1

00,028 REM DATA 1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087 '128+16*2 to 128+16*3-1

00,029 REM

00,030 REM DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488 '128+16*3 to 128+16*4-1

00,031 REM DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575 '128+16*4 to 128+16*5-1

00,032 REM DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600 '128+16*5 to 128+16*6-1

00,033 REM

00,034 REM DATA 1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103 '128+16*6 to 128+16*7-1 <-/

00,035 REM 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255

00,036 REM DATA 8216,8217,8218,8219,8220,8221,8222,8223,176 ,1118,1025,1105,171 ,187 ,175 ,8230 '128+16*7 to 128+16*8-1

00,037

00,038 'We need to replace 1251 only with the last line, above.

00,039

00,040 _DEFINE A-Z AS _INTEGER64

00,041

00,042 DIM byt AS STRING * 1

00,043 DIM byt2 AS STRING * 1

00,044 DIM byt3 AS STRING * 1

00,045 IF COMMAND\$ = "" THEN PRINT "Usage: UTF8toGesch.exe filename": SYSTEM

00,046 OPEN COMMAND\$ FOR BINARY AS #1

00,047 OPEN COMMAND\$ + ".Gesch" FOR BINARY AS #2

00,048 qq = LOF(1)

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,049 'PRINT qq
00,050 'Grrr, below not the same as ? qq i.e. not as it should?!
00,051 'PRINT "Converting"; LOF(1); "bytes..."
00,052 PRINT "Converting"; qq; "bytes..."
00,053 LOCATE CSRLIN, 1: PRINT "Done"; (0 * 100) \ qq; "%";
00,054 FOR iii = 1 TO qq
00,055     IF (iii * 100) \ qq > ((iii - 1) * 100) \ qq THEN LOCATE CSRLIN, 1: PRINT "Done"; (iii * 100) \ qq; "%";
00,056     GET #1, iii, byt$
00,057     SELECT CASE ASC(byt$)
00,058         CASE &HE2:
00,059             byt2$ = CHR$(32)
00,060             byt3$ = CHR$(32)
00,061             IF iii + 2 <= qq THEN
00,062                 GET #1, , byt2$
00,063                 GET #1, , byt3$
00,064             END IF
00,065             IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H9E THEN ' ,, lower " curved to left downwards
00,066                 iii = iii + 2
00,067                 bytNEW$ = CHR$(246)
00,068                 PUT #2, , bytNEW$
00,069             END IF
00,070             IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H9F THEN ' `` upper " curved to right downwards
00,071                 iii = iii + 2
00,072                 bytNEW$ = CHR$(247)
00,073                 PUT #2, , bytNEW$
00,074             END IF
00,075             IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H9C THEN ' upper " curved to right upwards
00,076                 iii = iii + 2
00,077                 bytNEW$ = CHR$(244)
00,078                 PUT #2, , bytNEW$
00,079             END IF
00,080             IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H9D THEN ' upper " curved to left downwards
00,081                 iii = iii + 2
00,082                 bytNEW$ = CHR$(245)
00,083                 PUT #2, , bytNEW$
00,084             END IF
00,085             IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H98 THEN ' upper ' curved to right upwards
00,086                 iii = iii + 2
00,087                 bytNEW$ = CHR$(240)
00,088                 PUT #2, , bytNEW$
00,089             END IF
00,090             IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H99 THEN ' upper ' curved to left downwards
00,091                 iii = iii + 2
00,092                 bytNEW$ = CHR$(241)
00,093                 PUT #2, , bytNEW$
00,094             END IF
00,095             IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H9A THEN ' lower ' curved to left downwards
00,096                 iii = iii + 2
00,097                 bytNEW$ = CHR$(242)

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,098      PUT #2, , bytNEW$
00,099      END IF
00,100      IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H9B THEN ' upper ' curved to right downwards
00,101          iii = iii + 2
00,102          bytNEW$ = CHR$(243)
00,103          PUT #2, , bytNEW$
00,104      END IF
00,105      IF ASC(byt2$) = &H80 AND (ASC(byt3$) = &H93 OR ASC(byt3$) = &H94) THEN ' -
00,106          iii = iii + 2
00,107          bytNEW$ = CHR$(196)
00,108          PUT #2, , bytNEW$
00,109      END IF
00,110      IF ASC(byt2$) = &H80 AND ASC(byt3$) = &HA6 THEN ' ...
00,111          iii = iii + 2
00,112          bytNEW$ = CHR$(255)
00,113          PUT #2, , bytNEW$
00,114      END IF
00,115      IF ASC(byt2$) = &H80 AND ASC(byt3$) = &H97 THEN ' double underscore
00,116          iii = iii + 2
00,117          bytNEW$ = CHR$(208)
00,118          PUT #2, , bytNEW$
00,119      END IF
00,120      IF ASC(byt2$) = &H96 AND ASC(byt3$) = &HAA THEN ' filled box
00,121          iii = iii + 2
00,122          bytNEW$ = CHR$(209)
00,123          PUT #2, , bytNEW$
00,124      END IF
00,125      IF ASC(byt2$) = &H96 AND ASC(byt3$) = &HAB THEN ' empty box
00,126          iii = iii + 2
00,127          bytNEW$ = CHR$(210)
00,128          PUT #2, , bytNEW$
00,129      END IF
00,130      IF ASC(byt2$) = &H96 AND ASC(byt3$) = &H80 THEN ' upper box
00,131          iii = iii + 2
00,132          bytNEW$ = CHR$(28)
00,133          PUT #2, , bytNEW$
00,134      END IF
00,135      IF ASC(byt2$) = &H96 AND ASC(byt3$) = &H81 THEN ' bold cursor
00,136          iii = iii + 2
00,137          bytNEW$ = CHR$(29)
00,138          PUT #2, , bytNEW$
00,139      END IF
00,140      IF ASC(byt2$) = &H96 AND ASC(byt3$) = &H84 THEN ' lower box
00,141          iii = iii + 2
00,142          bytNEW$ = CHR$(30)
00,143          PUT #2, , bytNEW$
00,144      END IF
00,145      IF ASC(byt2$) = &H8C AND ASC(byt3$) = &HA0 THEN ' high integral
00,146          iii = iii + 2

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,147      bytNEW$ = CHR$(209)
00,148      PUT #2, , bytNEW$
00,149      END IF
00,150      IF ASC(byt2$) = &H8C AND ASC(byt3$) = &HA1 THEN ' low integral
00,151          iii = iii + 2
00,152          bytNEW$ = CHR$(210)
00,153          PUT #2, , bytNEW$
00,154      END IF
00,155      IF ASC(byt2$) = &H89 AND ASC(byt3$) = &H88 THEN ' almost equal
00,156          iii = iii + 2
00,157          bytNEW$ = CHR$(127)
00,158          PUT #2, , bytNEW$
00,159      END IF
00,160      IF ASC(byt2$) = &H82 AND ASC(byt3$) = &H8A THEN ' under +
00,161          iii = iii + 2
00,162          bytNEW$ = CHR$(11)
00,163          PUT #2, , bytNEW$
00,164      END IF
00,165      IF ASC(byt2$) = &H82 AND ASC(byt3$) = &H8B THEN ' under -
00,166          iii = iii + 2
00,167          bytNEW$ = CHR$(12)
00,168          PUT #2, , bytNEW$
00,169      END IF
00,170
00,171      CASE &HC2:
00,172          byt2$ = CHR$(32)
00,173          IF iii + 1 <= qq THEN
00,174              GET #1, , byt2$
00,175          END IF
00,176          IF ASC(byt2$) = &HA0 THEN 'space
00,177              iii = iii + 1
00,178              bytNEW$ = CHR$(32)
00,179              PUT #2, , bytNEW$
00,180          END IF
00,181          IF ASC(byt2$) = &HAB THEN '<<
00,182              iii = iii + 1
00,183              bytNEW$ = CHR$(252)
00,184              PUT #2, , bytNEW$
00,185          END IF
00,186          IF ASC(byt2$) = &HBB THEN '>>
00,187              iii = iii + 1
00,188              bytNEW$ = CHR$(253)
00,189              PUT #2, , bytNEW$
00,190          END IF
00,191          IF ASC(byt2$) = &HB5 THEN 'micro
00,192              iii = iii + 1
00,193              bytNEW$ = CHR$(249)
00,194              PUT #2, , bytNEW$
00,195          END IF

```

```

00,196         IF ASC(byt2$) = &HA1 THEN 'r!
00,197             iii = iii + 1
00,198             bytNEW$ = CHR$(211)
00,199             PUT #2, , bytNEW$
00,200         END IF
00,201         IF ASC(byt2$) = &HBF THEN 'r?
00,202             iii = iii + 1
00,203             bytNEW$ = CHR$(212)
00,204             PUT #2, , bytNEW$
00,205         END IF
00,206
00,207
00,208     CASE &HC3:
00,209         byt2$ = CHR$(32)
00,210         IF iii + 1 <= qq THEN
00,211             GET #1, , byt2$
00,212         END IF
00,213         IF ASC(byt2$) = &H84 THEN 'A:
00,214             iii = iii + 1
00,215             bytNEW$ = CHR$(0)
00,216             PUT #2, , bytNEW$
00,217         END IF
00,218         IF ASC(byt2$) = &HA4 THEN 'a:
00,219             iii = iii + 1
00,220             bytNEW$ = CHR$(1)
00,221             PUT #2, , bytNEW$
00,222         END IF
00,223         IF ASC(byt2$) = &H96 THEN '0:
00,224             iii = iii + 1
00,225             bytNEW$ = CHR$(2)
00,226             PUT #2, , bytNEW$
00,227         END IF
00,228         IF ASC(byt2$) = &HB6 THEN 'o:
00,229             iii = iii + 1
00,230             bytNEW$ = CHR$(3)
00,231             PUT #2, , bytNEW$
00,232         END IF
00,233         IF ASC(byt2$) = &H9C THEN 'U:
00,234             iii = iii + 1
00,235             bytNEW$ = CHR$(4)
00,236             PUT #2, , bytNEW$
00,237         END IF
00,238         IF ASC(byt2$) = &HBC THEN 'u:
00,239             iii = iii + 1
00,240             bytNEW$ = CHR$(5)
00,241             PUT #2, , bytNEW$
00,242         END IF
00,243         IF ASC(byt2$) = &H9F THEN 'ss
00,244             iii = iii + 1

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,245      bytNEW$ = CHR$(6)
00,246      PUT #2, , bytNEW$
00,247      END IF
00,248      IF ASC(byt2$) = &H87 THEN 'C,
00,249          iii = iii + 1
00,250          bytNEW$ = CHR$(14)
00,251          PUT #2, , bytNEW$
00,252      END IF
00,253      IF ASC(byt2$) = &HA7 THEN 'c,
00,254          iii = iii + 1
00,255          bytNEW$ = CHR$(15)
00,256          PUT #2, , bytNEW$
00,257      END IF
00,258      IF ASC(byt2$) = &H82 THEN 'A^
00,259          iii = iii + 1
00,260          bytNEW$ = CHR$(198)
00,261          PUT #2, , bytNEW$
00,262      END IF
00,263      IF ASC(byt2$) = &HA2 THEN 'a^
00,264          iii = iii + 1
00,265          bytNEW$ = CHR$(199)
00,266          PUT #2, , bytNEW$
00,267      END IF
00,268      IF ASC(byt2$) = &H8A THEN 'E^
00,269          iii = iii + 1
00,270          bytNEW$ = CHR$(200)
00,271          PUT #2, , bytNEW$
00,272      END IF
00,273      IF ASC(byt2$) = &HAA THEN 'e^
00,274          iii = iii + 1
00,275          bytNEW$ = CHR$(201)
00,276          PUT #2, , bytNEW$
00,277      END IF
00,278      IF ASC(byt2$) = &H8E THEN 'I^
00,279          iii = iii + 1
00,280          bytNEW$ = CHR$(202)
00,281          PUT #2, , bytNEW$
00,282      END IF
00,283      IF ASC(byt2$) = &HAE THEN 'i^
00,284          iii = iii + 1
00,285          bytNEW$ = CHR$(203)
00,286          PUT #2, , bytNEW$
00,287      END IF
00,288      IF ASC(byt2$) = &H94 THEN 'O^
00,289          iii = iii + 1
00,290          bytNEW$ = CHR$(204)
00,291          PUT #2, , bytNEW$
00,292      END IF
00,293      IF ASC(byt2$) = &HB4 THEN 'o^

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,294      iii = iii + 1
00,295      bytNEW$ = CHR$(205)
00,296      PUT #2, , bytNEW$
00,297      END IF
00,298      IF ASC(byt2$) = &H9B THEN 'U^
00,299      iii = iii + 1
00,300      bytNEW$ = CHR$(206)
00,301      PUT #2, , bytNEW$
00,302      END IF
00,303      IF ASC(byt2$) = &HRB THEN 'u^
00,304      iii = iii + 1
00,305      bytNEW$ = CHR$(207)
00,306      PUT #2, , bytNEW$
00,307      END IF
00,308      IF ASC(byt2$) = &H89 THEN 'E'
00,309      iii = iii + 1
00,310      bytNEW$ = CHR$(16)
00,311      PUT #2, , bytNEW$
00,312      END IF
00,313      IF ASC(byt2$) = &HA9 THEN 'e'
00,314      iii = iii + 1
00,315      bytNEW$ = CHR$(17)
00,316      PUT #2, , bytNEW$
00,317      END IF
00,318      IF ASC(byt2$) = &H80 THEN 'A'
00,319      iii = iii + 1
00,320      bytNEW$ = CHR$(18)
00,321      PUT #2, , bytNEW$
00,322      END IF
00,323      IF ASC(byt2$) = &HA0 THEN 'a'
00,324      iii = iii + 1
00,325      bytNEW$ = CHR$(19)
00,326      PUT #2, , bytNEW$
00,327      END IF
00,328      IF ASC(byt2$) = &H88 THEN 'E'
00,329      iii = iii + 1
00,330      bytNEW$ = CHR$(20)
00,331      PUT #2, , bytNEW$
00,332      END IF
00,333      IF ASC(byt2$) = &HA8 THEN 'e'
00,334      iii = iii + 1
00,335      bytNEW$ = CHR$(21)
00,336      PUT #2, , bytNEW$
00,337      END IF
00,338      IF ASC(byt2$) = &H8C THEN 'I'
00,339      iii = iii + 1
00,340      bytNEW$ = CHR$(22)
00,341      PUT #2, , bytNEW$
00,342      END IF

```

```

00,343     IF ASC(byt2$) = &HAC THEN 'i'
00,344         iii = iii + 1
00,345         bytNEW$ = CHR$(23)
00,346         PUT #2, , bytNEW$
00,347     END IF
00,348     IF ASC(byt2$) = &H92 THEN 'O'
00,349         iii = iii + 1
00,350         bytNEW$ = CHR$(24)
00,351         PUT #2, , bytNEW$
00,352     END IF
00,353     IF ASC(byt2$) = &HB2 THEN 'o'
00,354         iii = iii + 1
00,355         bytNEW$ = CHR$(25)
00,356         PUT #2, , bytNEW$
00,357     END IF
00,358     IF ASC(byt2$) = &H99 THEN 'U'
00,359         iii = iii + 1
00,360         bytNEW$ = CHR$(26)
00,361         PUT #2, , bytNEW$
00,362     END IF
00,363     IF ASC(byt2$) = &HB9 THEN 'u'
00,364         iii = iii + 1
00,365         bytNEW$ = CHR$(27)
00,366         PUT #2, , bytNEW$
00,367     END IF
00,368     IF ASC(byt2$) = &HB7 THEN 'division'
00,369         iii = iii + 1
00,370         bytNEW$ = CHR$(31)
00,371         PUT #2, , bytNEW$
00,372     END IF
00,373     IF ASC(byt2$) = &H8F THEN 'I:
00,374         iii = iii + 1
00,375         bytNEW$ = CHR$(189)
00,376         PUT #2, , bytNEW$
00,377     END IF
00,378     IF ASC(byt2$) = &HAF THEN 'i:
00,379         iii = iii + 1
00,380         bytNEW$ = CHR$(190)
00,381         PUT #2, , bytNEW$
00,382     END IF
00,383     IF ASC(byt2$) = &H91 THEN 'N'
00,384         iii = iii + 1
00,385         bytNEW$ = CHR$(213)
00,386         PUT #2, , bytNEW$
00,387     END IF
00,388     IF ASC(byt2$) = &HB1 THEN 'n~
00,389         iii = iii + 1
00,390         bytNEW$ = CHR$(214)
00,391         PUT #2, , bytNEW$

```

```

00,392      END IF
00,393      IF ASC(byt2$) = &H9D THEN 'Y'
00,394          iii = iii + 1
00,395          bytNEW$ = CHR$(215)
00,396          PUT #2, , bytNEW$
00,397      END IF
00,398      IF ASC(byt2$) = &HBD THEN 'y'
00,399          iii = iii + 1
00,400          bytNEW$ = CHR$(216)
00,401          PUT #2, , bytNEW$
00,402      END IF
00,403      IF ASC(byt2$) = &H86 THEN 'AE'
00,404          iii = iii + 1
00,405          bytNEW$ = CHR$(220)
00,406          PUT #2, , bytNEW$
00,407      END IF
00,408      IF ASC(byt2$) = &HA6 THEN 'ae'
00,409          iii = iii + 1
00,410          bytNEW$ = CHR$(221)
00,411          PUT #2, , bytNEW$
00,412      END IF
00,413      IF ASC(byt2$) = &H81 THEN 'A'
00,414          iii = iii + 1
00,415          bytNEW$ = CHR$(7)
00,416          PUT #2, , bytNEW$
00,417      END IF
00,418      IF ASC(byt2$) = &HA1 THEN 'a'
00,419          iii = iii + 1
00,420          bytNEW$ = CHR$(8)
00,421          PUT #2, , bytNEW$
00,422      END IF
00,423      IF ASC(byt2$) = &H93 THEN 'O'
00,424          iii = iii + 1
00,425          bytNEW$ = CHR$(11)
00,426          PUT #2, , bytNEW$
00,427      END IF
00,428      IF ASC(byt2$) = &HB3 THEN 'o'
00,429          iii = iii + 1
00,430          bytNEW$ = CHR$(12)
00,431          PUT #2, , bytNEW$
00,432      END IF
00,433      IF ASC(byt2$) = &H8D THEN 'I'
00,434          iii = iii + 1
00,435          bytNEW$ = CHR$(28)
00,436          PUT #2, , bytNEW$
00,437      END IF
00,438      IF ASC(byt2$) = &HAD THEN 'i'
00,439          iii = iii + 1
00,440          bytNEW$ = CHR$(29)

```

```

00,441         PUT #2, , bytNEW$
00,442     END IF
00,443     IF ASC(byt2$) = &H9A THEN 'U'
00,444         iii = iii + 1
00,445         bytNEW$ = CHR$(30)
00,446         PUT #2, , bytNEW$
00,447     END IF
00,448     IF ASC(byt2$) = &HBA THEN 'u'
00,449         iii = iii + 1
00,450         bytNEW$ = CHR$(31)
00,451         PUT #2, , bytNEW$
00,452     END IF
00,453
00,454
00,455     CASE &HC4:
00,456         byt2$ = CHR$(32)
00,457         IF iii + 1 <= qq THEN
00,458             GET #1, , byt2$
00,459         END IF
00,460         IF ASC(byt2$) = &H82 THEN 'A kratko
00,461             iii = iii + 1
00,462             bytNEW$ = CHR$(181)
00,463             PUT #2, , bytNEW$
00,464         END IF
00,465         IF ASC(byt2$) = &H83 THEN 'a kratko
00,466             iii = iii + 1
00,467             bytNEW$ = CHR$(182)
00,468             PUT #2, , bytNEW$
00,469         END IF
00,470         IF ASC(byt2$) = &H94 THEN 'E kratko
00,471             iii = iii + 1
00,472             bytNEW$ = CHR$(183)
00,473             PUT #2, , bytNEW$
00,474         END IF
00,475         IF ASC(byt2$) = &H95 THEN 'e kratko
00,476             iii = iii + 1
00,477             bytNEW$ = CHR$(184)
00,478             PUT #2, , bytNEW$
00,479         END IF
00,480
00,481
00,482     CASE &HC5:
00,483         byt2$ = CHR$(32)
00,484         IF iii + 1 <= qq THEN
00,485             GET #1, , byt2$
00,486         END IF
00,487         IF ASC(byt2$) = &H92 THEN 'OE
00,488             iii = iii + 1
00,489             bytNEW$ = CHR$(222)

```

```

00,490         PUT #2, , bytNEW$
00,491     END IF
00,492     IF ASC(byt2$) = &H93 THEN 'oe
00,493         iii = iii + 1
00,494         bytNEW$ = CHR$(223)
00,495         PUT #2, , bytNEW$
00,496     END IF
00,497     IF ASC(byt2$) = &H8E THEN 'O kratko
00,498         iii = iii + 1
00,499         bytNEW$ = CHR$(185)
00,500         PUT #2, , bytNEW$
00,501     END IF
00,502     IF ASC(byt2$) = &H8F THEN 'o kratko
00,503         iii = iii + 1
00,504         bytNEW$ = CHR$(186)
00,505         PUT #2, , bytNEW$
00,506     END IF
00,507
00,508
00,509     CASE &HD0:
00,510         byt2$ = CHR$(32)
00,511         IF iii + 1 <= qq THEN
00,512             GET #1, , byt2$
00,513         END IF
00,514         IF ASC(byt2$) = &H81 THEN 'E:
00,515             iii = iii + 1
00,516             bytNEW$ = CHR$(250)
00,517             PUT #2, , bytNEW$
00,518         END IF
00,519         IF ASC(byt2$) = &H8E THEN 'Y kratko
00,520             iii = iii + 1
00,521             bytNEW$ = CHR$(187)
00,522             PUT #2, , bytNEW$
00,523         END IF
00,524         IF ASC(byt2$) >= &H90 AND ASC(byt2$) <= &HBF THEN
00,525             iii = iii + 1
00,526             bytNEW$ = CHR$(128 + (ASC(byt2$) - &H90))
00,527             PUT #2, , bytNEW$
00,528         END IF
00,529
00,530     CASE &HD1:
00,531         byt2$ = CHR$(32)
00,532         IF iii + 1 <= qq THEN
00,533             GET #1, , byt2$
00,534         END IF
00,535         IF ASC(byt2$) = &H91 THEN 'e:
00,536             iii = iii + 1
00,537             bytNEW$ = CHR$(251)
00,538             PUT #2, , bytNEW$

```

```

00,539      END IF
00,540      IF ASC(byt2$) = &H9E THEN 'y(
00,541          iii = iii + 1
00,542          bytNEW$ = CHR$(188)
00,543          PUT #2, , bytNEW$
00,544      END IF
00,545      IF ASC(byt2$) >= &H80 AND ASC(byt2$) <= &H8F THEN 'A..ya
00,546          iii = iii + 1
00,547          bytNEW$ = CHR$(224 + (ASC(byt2$) - &H80))
00,548          PUT #2, , bytNEW$
00,549      END IF
00,550
00,551      CASE ELSE
00,552          PUT #2, , byt$
00,553  END SELECT
00,554
00,555
00,556  'SELECT CASE ASC(byt$)
00,557  '    CASE 130:
00,558  '        bytNEW$ = CHR$(242)
00,559  '        PUT #2, , bytNEW$
00,560
00,561  '    CASE 132:
00,562  '        bytNEW$ = CHR$(246)
00,563  '        PUT #2, , bytNEW$
00,564
00,565  '    CASE 133:
00,566  '        bytNEW$ = CHR$(255)
00,567  '        PUT #2, , bytNEW$
00,568
00,569  '    CASE 145:
00,570  '        bytNEW$ = CHR$(240)
00,571  '        PUT #2, , bytNEW$
00,572
00,573  '    CASE 146:
00,574  '        bytNEW$ = CHR$(241)
00,575  '        PUT #2, , bytNEW$
00,576
00,577  '    CASE 147:
00,578  '        bytNEW$ = CHR$(244)
00,579  '        PUT #2, , bytNEW$
00,580
00,581  '    CASE 148:
00,582  '        bytNEW$ = CHR$(245)
00,583  '        PUT #2, , bytNEW$
00,584
00,585  '    CASE 152:
00,586  '        bytNEW$ = CHR$(0)
00,587  '        PUT #2, , bytNEW$

```

```

00,588
00,589 ' CASE 160:
00,590 '     bytNEW$ = CHR$(0)
00,591 '     PUT #2, , bytNEW$
00,592
00,593 ' CASE 162:
00,594 '     bytNEW$ = CHR$(249)
00,595 '     PUT #2, , bytNEW$
00,596
00,597 ' CASE 168:
00,598 '     bytNEW$ = CHR$(250)
00,599 '     PUT #2, , bytNEW$
00,600
00,601 ' CASE 184:
00,602 '     bytNEW$ = CHR$(251)
00,603 '     PUT #2, , bytNEW$
00,604
00,605 '     'Not in the tables above, but should be changed [
00,606 ' CASE 171:
00,607 '     bytNEW$ = CHR$(252)
00,608 '     PUT #2, , bytNEW$
00,609 ' CASE 187:
00,610 '     bytNEW$ = CHR$(253)
00,611 '     PUT #2, , bytNEW$
00,612 ' CASE 150:
00,613 '     bytNEW$ = CHR$(196)
00,614 '     PUT #2, , bytNEW$
00,615 ' CASE 151:
00,616 '     bytNEW$ = CHR$(196)
00,617 '     PUT #2, , bytNEW$
00,618 ' CASE 173:
00,619 '     bytNEW$ = CHR$(196)
00,620 '     PUT #2, , bytNEW$
00,621 '     'Not in the tables above, but should be changed ]
00,622
00,623 ' CASE 128 + 16 * 4 TO 128 + 16 * 5 - 1:
00,624 '     bytNEW$ = CHR$(ASC(byt$) - (4 - 0) * 16)
00,625 '     PUT #2, , bytNEW$
00,626 ' CASE 128 + 16 * 5 TO 128 + 16 * 6 - 1:
00,627 '     bytNEW$ = CHR$(ASC(byt$) - (5 - 1) * 16)
00,628 '     PUT #2, , bytNEW$
00,629 ' CASE 128 + 16 * 6 TO 128 + 16 * 7 - 1:
00,630 '     bytNEW$ = CHR$(ASC(byt$) - (6 - 2) * 16)
00,631 '     PUT #2, , bytNEW$
00,632 ' CASE 128 + 16 * 7 TO 128 + 16 * 8 - 1:
00,633 '     bytNEW$ = CHR$(ASC(byt$) - (7 - 6) * 16)
00,634 '     PUT #2, , bytNEW$
00,635 ' CASE ELSE:
00,636 '     PUT #2, , byt$

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```
00,637 'END SELECT
00,638 NEXT
00,639 LOCATE CSRLIN, 1: PRINT "Done"; (qq * 100) \ qq; "%"
00,640 CLOSE #1, #2
00,641 SYSTEM
00,642
00,643
00,644 'Microsoft_pc_cpGESCH: 'Schpitz/Gesch (a.k.a. Georgievica a.k.a. Geschovica) is Sanmayce's layout, combining the MIK and 437, in this way: (NOT RECODING only 009,010,013):
00,645 '' shpitz
00,646
00,647 '' DEFINITIONS
00,648 '' top or extreme part
00,649 '' the ultimate, the best of, "cool"
00,650 '' typical, classic, a perfect example of; definitively
00,651
00,652 '' LANGUAGES OF ORIGIN
00,653 '' Yiddish
00,654
00,655 '' ETYMOLOGY
00,656 '' ????? shpits 'tip, peak'
00,657
00,658 '' ALTERNATIVE SPELLINGS
00,659 '' schpitz, shpits, shpitz, shpitsy
00,660
00,661 ''NOTES
00,662 ''Steinmetz, "Yiddish and English" has an entry for shpits, but the meaning is simply 'tip.'
00,663
00,664 '' ASCII 000..031 have to accomodate German and French ' &H203e = 8254 is upperscore ' &H00DC=220
00,665 'DATA 196,228,214,246,&H00DC,&H00FC,223,&H00c1,&H00e1,&h00C7,&H00E7
00,666 ''A:a:O:o:U:u:ss A' a' C'c'
00,667 'DATA &H0c9,&H0e9,&H0c0,&H0e0,&H0c8,&H0e8,&H0cc,&H0ec,&H0d2,&H0f2,&H0d9,&H0f9
00,668 '' E'e' A'a'E'e'I'i'O'o'U'u'
00,669
00,670 ''DATA &H00ch,&H00eb,&H00cf,&H00ef
00,671 '' E:e: I:i: ' E:e: are present in Russian, so remove them
00,672
00,673 ''First half of big Cyrillic letters:
00,674 'DATA 1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055
00,675 ''Second half of big Cyrillic letters:
00,676 'DATA 1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071
00,677 ''First half of small Cyrillic letters:
00,678 'DATA 1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087
00,679 ''1st third of CP437 drawing symbols:
00,680 'DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488
00,681 ''2nd third of CP437 drawing symbols:
00,682 'DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575
00,683 ''3rd third of CP437 drawing symbols:
00,684 'DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600
00,685 ''Second half of small Cyrillic letters:
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```
00,686 'DATA 1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103
00,687 ''Last 16 of CP437 symbols:
00,688 ''                                u-kr E:  e:  << >> Up_ ...
00,689 ''DATA 8216,8217,8218,8219,8220,8221,8222,8223,176,1118,1025,1105,171,187,175,8230
00,690 ''                                MICRO E:  e:  << >> Up_ ...
00,691 'DATA 8216,8217,8218,8219,8220,8221,8222,8223,176,&h00b5,1025,1105,171,187,175,8230
00,692
00,693 'DATA &H00c6,&H00E6,&H0152,&H0153
00,694 '' AE ae OEoe
00,695
00,696 'DATA &h2248
00,697 ''almost equal to
00,698 ''DATA &H221a
00,699 ''square root
00,700
00,701 'DATA &H0102,&H0103,&H0114,&H0115,&H014e,&H014f,&H040e,&H045e,&H00cf,&H00ef
00,702 ''Cyrillic short vowels: AaEeOoYy I:i:
00,703 'DATA &H00c2,&H00e2,&H00ca,&H00ea,&H00ce,&H00ee,&H00d4,&H00f4,&H00db,&H00fb
00,704 '' A^a^ E^e^ I^i^ O^o^ U^u^
00,705
00,706 'DATA &h2017,&H2320,&H2321,&h00a1,&h00bf,&h00d1,&h00f1,&h00dd,&h00fd
00,707 ''doubleunderline IntegralH IntegralL, Spanish: r! r? N~ n~ Y~ y~
00,708
00,709 'DATA &h00cd,&h00ed,&h00da,&h00fa
00,710 '' I^i^ U^u^
00,711
00,712 'DATA &h00D3,&h00F3
00,713 '' O^ o^
00,714
00,715 '' Portuguese makes use of five diacritics: the cedilla (c,), acute accent (a' e' i' o' u'), circumflex accent (a^, e^, o^), tilde (a~, o~), and grave accent (a', and rarely e', i', o', and u').
00,716
```

1251toGesch.bas:

```
00,001 REM Converting windows_cp1251 to Gesch codepage
00,002
00,003 $SCREENHIDE
00,004 $CONSOLE
00,005 _CONSOLE ON
00,006 _CONSOLETITLE "Gesch codepage convertor"
00,007 _DEST _CONSOLE
00,008
00,009 REM Microsoft_windows_cp1251: 'Cyrillic alphabet such as Russian, Bulgarian, Serbian Cyrillic and other languages. It is the most widely used for encoding the Bulgarian, Serbian and Macedonian languages.
00,010 REM      128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
00,011 REM DATA 1026,1027,8218,1107,8222,8230,8224,8225,8364,8240,1033,8249,1034,1036,1035,1039 '128+16*0 to 128+16*1-1
00,012 REM      144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
00,013 REM DATA 1106,8216,8217,8220,8221,8226,8211,8212,0 ,8482,1113,8250,1114,1116,1115,1119 '128+16*1 to 128+16*2-1
00,014 REM      160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
00,015 REM DATA 160 ,1038,1118,1032,164 ,1168,166 ,167 ,1025,169 ,1028,171 ,172 ,173 ,174 ,1031 '128+16*2 to 128+16*3-1
00,016 REM      176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip
00,017 REM DATA 176 ,177 ,1030,1110,1169,181 ,182 ,183 ,1105,8470,1108,187 ,1112,1029,1109,1111 '128+16*3 to 128+16*4-1
00,018 REM Cyrillic:
00,019 REM DATA 1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055 '128+16*4 to 128+16*5-1
00,020 REM DATA 1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071 '128+16*5 to 128+16*6-1
00,021 REM DATA 1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087 '128+16*6 to 128+16*7-1
00,022 REM DATA 1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103 '128+16*7 to 128+16*8-1 --\
00,023 REM
00,024 REM Microsoft_pc_cpGESCH: 'Gesch is Sanmayce's layout, combining the MIK and 437, in this way:
00,025 REM
00,026 REM DATA 1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055 '128+16*0 to 128+16*1-1
00,027 REM DATA 1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071 '128+16*1 to 128+16*2-1
00,028 REM DATA 1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087 '128+16*2 to 128+16*3-1
00,029 REM
00,030 REM DATA 9617,9618,9619,9474,9508,9569,9570,9558,9557,9571,9553,9559,9565,9564,9563,9488 '128+16*3 to 128+16*4-1
00,031 REM DATA 9492,9524,9516,9500,9472,9532,9566,9567,9562,9556,9577,9574,9568,9552,9580,9575 '128+16*4 to 128+16*5-1
00,032 REM DATA 9576,9572,9573,9561,9560,9554,9555,9579,9578,9496,9484,9608,9604,9612,9616,9600 '128+16*5 to 128+16*6-1
00,033 REM
00,034 REM DATA 1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103 '128+16*6 to 128+16*7-1 <-/
00,035 REM      240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
00,036 REM DATA 8216,8217,8218,8219,8220,8221,8222,8223,176 ,1118,1025,1105,171 ,187 ,175 ,8230 '128+16*7 to 128+16*8-1
00,037
00,038 'We need to replace 1251 only with the last line, above.
00,039
00,040 _DEFINE A-Z AS _INTEGER64
00,041
00,042 DIM byt AS STRING * 1
00,043 IF COMMAND$ = "" THEN PRINT "Usage: 1251toGesch.exe filename": SYSTEM
00,044 OPEN COMMAND$ FOR BINARY AS #1
00,045 OPEN COMMAND$ + ".Gesch" FOR BINARY AS #2
00,046 qq = LOF(1)
00,047 'PRINT qq
00,048 'Grrr, below not the same as ? qq i.e. not as it should?!
00,049 'PRINT "Converting"; LOF(1); "bytes..."
00,050 PRINT "Converting"; qq; "bytes..."
00,051 FOR iii = 1 TO qq
00,052     GET #1, iii, byt$
00,053     SELECT CASE ASC(byt$)
00,054         CASE 130:
00,055             bytNEW$ = CHR$(242)
00,056             PUT #2, , bytNEW$
00,057
00,058         CASE 132:
00,059             bytNEW$ = CHR$(246)
00,060             PUT #2, , bytNEW$
00,061
00,062         CASE 133:
00,063             bytNEW$ = CHR$(255)
00,064             PUT #2, , bytNEW$
00,065

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```
00,066      CASE 145:
00,067          bytNEW$ = CHR$(240)
00,068          PUT #2, , bytNEW$
00,069
00,070      CASE 146:
00,071          bytNEW$ = CHR$(241)
00,072          PUT #2, , bytNEW$
00,073
00,074      CASE 147:
00,075          bytNEW$ = CHR$(244)
00,076          PUT #2, , bytNEW$
00,077
00,078      CASE 148:
00,079          bytNEW$ = CHR$(245)
00,080          PUT #2, , bytNEW$
00,081
00,082      CASE 152:
00,083          bytNEW$ = CHR$(0)
00,084          PUT #2, , bytNEW$
00,085
00,086      CASE 160:
00,087          bytNEW$ = CHR$(0)
00,088          PUT #2, , bytNEW$
00,089
00,090      CASE 162:
00,091          bytNEW$ = CHR$(249)
00,092          PUT #2, , bytNEW$
00,093
00,094      CASE 168:
00,095          bytNEW$ = CHR$(250)
00,096          PUT #2, , bytNEW$
00,097
00,098      CASE 184:
00,099          bytNEW$ = CHR$(251)
00,100          PUT #2, , bytNEW$
00,101
00,102          'Not in the tables above, but should be changed [
00,103      CASE 171:
00,104          bytNEW$ = CHR$(252)
00,105          PUT #2, , bytNEW$
00,106      CASE 187:
00,107          bytNEW$ = CHR$(253)
00,108          PUT #2, , bytNEW$
00,109      CASE 150:
00,110          bytNEW$ = CHR$(196)
00,111          PUT #2, , bytNEW$
00,112      CASE 151:
00,113          bytNEW$ = CHR$(196)
00,114          PUT #2, , bytNEW$
```

```

00,115      CASE 173:
00,116          bytNEW$ = CHR$(196)
00,117          PUT #2, , bytNEW$
00,118          'Not in the tables above, but should be changed ]
00,119
00,120      CASE 128 + 16 * 4 TO 128 + 16 * 5 - 1:
00,121          bytNEW$ = CHR$(ASC(byt$) - (4 - 0) * 16)
00,122          PUT #2, , bytNEW$
00,123      CASE 128 + 16 * 5 TO 128 + 16 * 6 - 1:
00,124          bytNEW$ = CHR$(ASC(byt$) - (5 - 1) * 16)
00,125          PUT #2, , bytNEW$
00,126      CASE 128 + 16 * 6 TO 128 + 16 * 7 - 1:
00,127          bytNEW$ = CHR$(ASC(byt$) - (6 - 2) * 16)
00,128          PUT #2, , bytNEW$
00,129      CASE 128 + 16 * 7 TO 128 + 16 * 8 - 1:
00,130          bytNEW$ = CHR$(ASC(byt$) - (7 - 6) * 16)
00,131          PUT #2, , bytNEW$
00,132      CASE ELSE:
00,133          PUT #2, , byt$
00,134      END SELECT
00,135 NEXT
00,136 CLOSE #1, #2
00,137 SYSTEM
00,138

```

MEM.H:

```

00,001 #include<windows.h>
00,002 #include<stdio.h>
00,003 #include<tchar.h>
00,004
00,005
00,006 uint64 MemInUsePercent();
00,007 uint64 TotalPhysicalMem ();
00,008 uint64 FreePhysicalMem ();
00,009 uint64 TotalPagingFile ();
00,010 uint64 FreePagingFile ();
00,011 uint64 TotalVirtualMem ();
00,012 uint64 FreeVirtualMem ();
00,013 uint64 FreeExtendedMem ();
00,014
00,015 static float CalculateCPULoad();
00,016 static unsigned long long FileTimeToInt64();
00,017 float GetCPULoad();
00,018
00,019
00,020 uint64 MemInUsePercent () {
00,021     MEMORYSTATUSEX statex;
00,022     statex.dwLength = sizeof (statex);
00,023     GlobalMemoryStatusEx (&statex);

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```
00,024 return statex.dwMemoryLoad;
00,025 }
00,026
00,027 uint64 TotalPhysicalMem () {
00,028     MEMORYSTATUSEX statex;
00,029     statex.dwLength = sizeof (statex);
00,030     GlobalMemoryStatusEx (&statex);
00,031     return statex.ullTotalPhys;
00,032 }
00,033
00,034 uint64 FreePhysicalMem () {
00,035     MEMORYSTATUSEX statex;
00,036     statex.dwLength = sizeof (statex);
00,037     GlobalMemoryStatusEx (&statex);
00,038     return statex.ullAvailPhys;
00,039 }
00,040
00,041 uint64 TotalPagingFile () {
00,042     MEMORYSTATUSEX statex;
00,043     statex.dwLength = sizeof (statex);
00,044     GlobalMemoryStatusEx (&statex);
00,045     return statex.ullTotalPageFile;
00,046 }
00,047
00,048 uint64 FreePagingFile () {
00,049     MEMORYSTATUSEX statex;
00,050     statex.dwLength = sizeof (statex);
00,051     GlobalMemoryStatusEx (&statex);
00,052     return statex.ullAvailPageFile;
00,053 }
00,054
00,055 uint64 TotalVirtualMem () {
00,056     MEMORYSTATUSEX statex;
00,057     statex.dwLength = sizeof (statex);
00,058     GlobalMemoryStatusEx (&statex);
00,059     return statex.ullTotalVirtual;
00,060 }
00,061
00,062 uint64 FreeVirtualMem () {
00,063     MEMORYSTATUSEX statex;
00,064     statex.dwLength = sizeof (statex);
00,065     GlobalMemoryStatusEx (&statex);
00,066     return statex.ullAvailVirtual;
00,067 }
00,068
00,069 uint64 FreeExtendedMem () {
00,070     MEMORYSTATUSEX statex;
00,071     statex.dwLength = sizeof (statex);
00,072     GlobalMemoryStatusEx (&statex);
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```
00,073 return statex.ullAvailExtendedVirtual;
00,074 }
00,075
00,076 static float CalculateCPULoad(unsigned long long idleTicks, unsigned long long totalTicks)
00,077 {
00,078     static unsigned long long _previousTotalTicks = 0;
00,079     static unsigned long long _previousIdleTicks = 0;
00,080
00,081     unsigned long long totalTicksSinceLastTime = totalTicks - _previousTotalTicks;
00,082     unsigned long long idleTicksSinceLastTime = idleTicks - _previousIdleTicks;
00,083
00,084
00,085     float ret = 1.0f - ((totalTicksSinceLastTime > 0) ? ((float)idleTicksSinceLastTime) / totalTicksSinceLastTime : 0);
00,086
00,087     _previousTotalTicks = totalTicks;
00,088     _previousIdleTicks = idleTicks;
00,089     return ret;
00,090 }
00,091
00,092 static unsigned long long FileTimeToInt64(const FILETIME & ft)
00,093 {
00,094     return (((unsigned long long)(ft.dwHighDateTime)) << 32) | ((unsigned long long)ft.dwLowDateTime);
00,095 }
00,096
00,097 // Returns 1.0f for "CPU fully pinned", 0.0f for "CPU idle", or somewhere in between
00,098 // You'll need to call this at regular intervals, since it measures the load between
00,099 // the previous call and the current one. Returns -1.0 on error.
00,100 float GetCPULoad()
00,101 {
00,102     FILETIME idleTime, kernelTime, userTime;
00,103     return GetSystemTimes(&idleTime, &kernelTime, &userTime) ? CalculateCPULoad(FileTimeToInt64(idleTime), FileTimeToInt64(kernelTime) + FileTimeToInt64(userTime)) : -1.0f;
00,104 }
```

XGRAM PAGODA5 64bit.bat:

```
00,001 @if '%1' == '' goto Usage
00,002 @if '%2' == '' goto Usage
00,003 @goto Doit
00,004 :Usage
00,005 @echo Usage: PAGODA5.bat lowercasedword corpusname
00,006 @goto Finish
00,007 :Doit
00,008
00,009 if exist %2.01.txt.sorted goto OK1
00,010 if not exist %2.01.txt.sorted.7z goto RIPanyway
00,011 7za x86_v1900.exe e -y %2.01.txt.sorted.7z
00,012 :OK1
00,013 if exist %2.02.txt.sorted goto OK2
00,014 if not exist %2.02.txt.sorted.7z goto RIPanyway
00,015 7za x86_v1900.exe e -y %2.02.txt.sorted.7z
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```

00,016 :OK2
00,017 if exist %2.03.txt.sorted goto OK3
00,018 if not exist %2.03.txt.sorted.7z goto RIPanyway
00,019 7za_x86_v1900.exe e -y %2.03.txt.sorted.7z
00,020 :OK3
00,021 if exist %2.04.txt.sorted goto OK4
00,022 if not exist %2.04.txt.sorted.7z goto RIPanyway
00,023 7za_x86_v1900.exe e -y %2.04.txt.sorted.7z
00,024 :OK4
00,025 if exist %2.05.txt.sorted goto OK5
00,026 if not exist %2.05.txt.sorted.7z goto RIPanyway
00,027 7za_x86_v1900.exe e -y %2.05.txt.sorted.7z
00,028 :OK5
00,029
00,030 goto SkipRIP
00,031
00,032 :RIPanyway
00,033
00,034 @dir %2/b >%2.lst
00,035
00,036 Leprechaun_x-leton_64bit_Intel_01_001p.exe %2.lst %2.01 14123456 Y
00,037 @if %errorlevel% == 0 goto OKay1
00,038 Leprechaun_x-leton_64bit_Intel_01_008p.exe %2.lst %2.01 14123456 Y
00,039 @if %errorlevel% == 0 goto OKay1
00,040 Leprechaun_x-leton_64bit_Intel_01_512p.exe %2.lst %2.01 14123456 Y
00,041 @if %errorlevel% == 0 goto OKay1
00,042 del %2.01
00,043 goto Finish
00,044 :OKay1
00,045
00,046 Leprechaun_x-leton_64bit_Intel_02_001p.exe %2.lst %2.02 14123456 Y
00,047 @if %errorlevel% == 0 goto OKay2
00,048 Leprechaun_x-leton_64bit_Intel_02_008p.exe %2.lst %2.02 14123456 Y
00,049 @if %errorlevel% == 0 goto OKay2
00,050 Leprechaun_x-leton_64bit_Intel_02_512p.exe %2.lst %2.02 14123456 Y
00,051 @if %errorlevel% == 0 goto OKay2
00,052 del %2.02
00,053 goto Finish
00,054 :OKay2
00,055
00,056 Leprechaun_x-leton_64bit_Intel_03_001p.exe %2.lst %2.03 14123456 Y
00,057 @if %errorlevel% == 0 goto OKay3
00,058 Leprechaun_x-leton_64bit_Intel_03_008p.exe %2.lst %2.03 14123456 Y
00,059 @if %errorlevel% == 0 goto OKay3
00,060 Leprechaun_x-leton_64bit_Intel_03_512p.exe %2.lst %2.03 14123456 Y
00,061 @if %errorlevel% == 0 goto OKay3
00,062 del %2.03
00,063 goto Finish
00,064 :OKay3

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

```

00,065
00,066 Leprechaun_x-leton_64bit_Intel_04_001p.exe %2.lst %2.04 14123456 Y
00,067 @if %errorlevel% == 0 goto Okay4
00,068 Leprechaun_x-leton_64bit_Intel_04_008p.exe %2.lst %2.04 14123456 Y
00,069 @if %errorlevel% == 0 goto Okay4
00,070 Leprechaun_x-leton_64bit_Intel_04_512p.exe %2.lst %2.04 14123456 Y
00,071 @if %errorlevel% == 0 goto Okay4
00,072 del %2.04
00,073 goto Finish
00,074 :Okay4
00,075
00,076 Leprechaun_x-leton_64bit_Intel_05_001p.exe %2.lst %2.05 14123456 Y
00,077 @if %errorlevel% == 0 goto Okay5
00,078 Leprechaun_x-leton_64bit_Intel_05_008p.exe %2.lst %2.05 14123456 Y
00,079 @if %errorlevel% == 0 goto Okay5
00,080 Leprechaun_x-leton_64bit_Intel_05_512p.exe %2.lst %2.05 14123456 Y
00,081 @if %errorlevel% == 0 goto Okay5
00,082 del %2.05
00,083 goto Finish
00,084 :Okay5
00,085
00,086 sort /+10 /M 1012012 %2.01 /O %2.01.txt.sorted
00,087 sort /+10 /M 1012012 %2.02 /O %2.02.txt.sorted
00,088 sort /+10 /M 1012012 %2.03 /O %2.03.txt.sorted
00,089 sort /+10 /M 1012012 %2.04 /O %2.04.txt.sorted
00,090 sort /+10 /M 1012012 %2.05 /O %2.05.txt.sorted
00,091
00,092 @rem To avoid 'C:\Users\GOLDEN~1\AppData\Local\Temp\srt6CD8.tmp There is not enough space on the disk.' failures:
00,093 @rem set TMP=d:\
00,094 @rem set TEMP=d:\
00,095 @rem Of course, SSD if possible.
00,096
00,097 if exist %2.01.txt.sorted.7z del %2.01.txt.sorted.7z
00,098 if exist %2.02.txt.sorted.7z del %2.02.txt.sorted.7z
00,099 if exist %2.03.txt.sorted.7z del %2.03.txt.sorted.7z
00,100 if exist %2.04.txt.sorted.7z del %2.04.txt.sorted.7z
00,101 if exist %2.05.txt.sorted.7z del %2.05.txt.sorted.7z
00,102
00,103 7za_x86_v1900.exe a %2.01.txt.sorted.7z %2.01.txt.sorted
00,104 7za_x86_v1900.exe a %2.02.txt.sorted.7z %2.02.txt.sorted
00,105 7za_x86_v1900.exe a %2.03.txt.sorted.7z %2.03.txt.sorted
00,106 7za_x86_v1900.exe a %2.04.txt.sorted.7z %2.04.txt.sorted
00,107 7za_x86_v1900.exe a %2.05.txt.sorted.7z %2.05.txt.sorted
00,108
00,109 @del %2.lst
00,110 @del %2.01
00,111 @del %2.02
00,112 @del %2.03
00,113 @del %2.04

```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Yasakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

```
00,114 @del %2.05
00,115 @del Leprechaun.LOG
00,116
00,117 @echo.
00,118 @dir %2.07.txt.sorted*
00,119
00,120 :SkipRIP
00,121
00,122 copy %2.01.txt.sorted Gallowwalker.1.txt.sorted /y
00,123 copy %2.02.txt.sorted Gallowwalker.2.txt.sorted /y
00,124 copy %2.03.txt.sorted Gallowwalker.3.txt.sorted /y
00,125 copy %2.04.txt.sorted Gallowwalker.4.txt.sorted /y
00,126 copy %2.05.txt.sorted Gallowwalker.5.txt.sorted /y
00,127
00,128 Kazahana_Hexadecad_GCC_102_32bit.exe "%1" Gallowwalker.1.txt.sorted 1023
00,129 sort /R Kazahana.txt /O Kazahana_%1.1-1.txt
00,130 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_" Gallowwalker.2.txt.sorted 1023
00,131 sort /R Kazahana.txt /O Kazahana_%1.2-1.txt
00,132 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_" Gallowwalker.2.txt.sorted 1023
00,133 sort /R Kazahana.txt /O Kazahana_%1.2-2.txt
00,134 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.3.txt.sorted 1023
00,135 sort /R Kazahana.txt /O Kazahana_%1.3-1.txt
00,136 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_" Gallowwalker.3.txt.sorted 1023
00,137 sort /R Kazahana.txt /O Kazahana_%1.3-2.txt
00,138 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.3.txt.sorted 1023
00,139 sort /R Kazahana.txt /O Kazahana_%1.3-3.txt
00,140 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.4.txt.sorted 1023
00,141 sort /R Kazahana.txt /O Kazahana_%1.4-1.txt
00,142 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.4.txt.sorted 1023
00,143 sort /R Kazahana.txt /O Kazahana_%1.4-2.txt
00,144 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.4.txt.sorted 1023
00,145 sort /R Kazahana.txt /O Kazahana_%1.4-3.txt
00,146 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.4.txt.sorted 1023
00,147 sort /R Kazahana.txt /O Kazahana_%1.4-4.txt
00,148 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.5.txt.sorted 1023
00,149 sort /R Kazahana.txt /O Kazahana_%1.5-1.txt
00,150 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.5.txt.sorted 1023
00,151 sort /R Kazahana.txt /O Kazahana_%1.5-2.txt
00,152 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.5.txt.sorted 1023
00,153 sort /R Kazahana.txt /O Kazahana_%1.5-3.txt
00,154 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.5.txt.sorted 1023
00,155 sort /R Kazahana.txt /O Kazahana_%1.5-4.txt
00,156 Kazahana_Hexadecad_GCC_102_32bit.exe "%1_." Gallowwalker.5.txt.sorted 1023
00,157 sort /R Kazahana.txt /O Kazahana_%1.5-5.txt
00,158
00,159 dir Kazahana_%1.*.txt/b/on>q
00,160 LineJustify_PAGODA05.exe q
00,161 copy Kazahana_%1.*.txt.PAD %2_%1.PAGODA-order-5.txt/b/y
00,162
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/Masakari.zip

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip

```
00,163 Leprechaun_x-leton_64bit_Intel_01_001p.exe q q.wrd 14123456 Y
00,164 @if %errorlevel% == 0 goto OKwrd
00,165 Leprechaun_x-leton_64bit_Intel_01_008p.exe q q.wrd 14123456 Y
00,166 @if %errorlevel% == 0 goto OKwrd
00,167 Leprechaun_x-leton_64bit_Intel_01_512p.exe q q.wrd 14123456 Y
00,168 :OKwrd
00,169
00,170 sort /R q.wrd /O %2_%1.PAGODA-order-5.wrd
00,171
00,172 @del q
00,173 @del *.PAD
00,174 @del q.wrd
00,175 @del Kazahana.txt
00,176 @del Leprechaun.LOG
00,177 @del Kazahana_%1.*.txt
00,178
00,179 :Finish
```

Listing: MASAKARI_Vanilla.BAS (r.8.1+); Last version: 2021-Jul-28; Font: MxPlus_ToshibaTxl2_8x16.ttf; Downloadable at: www.sanmayce.com/yasakari.zip