```
                        ORG   :EBF4
                *
                *
                *
                *   =====================
                *** SCREEN EDITOR PACKAGE ***
                *   =====================
                *
                *
                *********************
                * INITIALISE EDITOR *
                *********************
                *
                * Sets up a mode 0 screen, clears 00A8-00AD, 00AF.
                * Prints text from 00A2 onwards; cursor at top left.
                *
                * Exit: HL: Cursor position (top left).
                *       AF corrupted, BCDE preserved.
                *
EBF4 3EFF       EINIT   MVI   A,:FF      Init mode 0
EBF6 CDD9E3             CALL  :E3D9      Change to mode 0
EBF9 3E0C               MVI   A,:0C
EBFB CD02E1             CALL  :E102      Clear screen
EBFE 210000             LXI   H,:0000    )
EC01 22A900             SHLD  :00A9      )
EC04 22AC00             SHLD  :00AC      )
EC07 7C                 MOV   A,H        ) Clear edit pointers
EC08 32A800             STA   :00A8      )
EC0B 32AB00             STA   :00AB      )
EC0E 32AF00             STA   :00AF      )
EC11 2A7200             LHLD  :0072      Get cursor pos addr
EC14 CD6BE3             CALL  :E36B      Delete cursor
EC17 CD3AEF             CALL  :EF3A      Print full screen from
                                         start of edit buffer
EC1A CD30E3             CALL  :E330      Put cursor on screen
EC1D C9                 RET
                *
                *************
                * OBEY EDIT *
                *************
                *
                * A given character is inserted in the buffer,
                * except:  #08  deletes a character
                *          #10  moves cursor up
                *          #11  moves cursor down
                *          #12  moves cursor left
                *          #13  moves cursor right
                *          #14  moves window up
                *          #15  moves window down
                *          #16  moves window left
                *          #17  moves window right
                *
                * Entry: A:    Character given.
                * Exit:   CY=1: If done.
                *         CY=0: Not done.
                *         ABCDEHL + rest of F preserved.
                *
EC1E FE08       EOBEY   CPI   :08
EC20 CACCEF             JZ    :EFCC      Delete character
EC23 FE10               CPI   :10
EC25 DA4BEF             JC    :EF4B      Insert character
EC28 CA88ED             JZ    :ED88      Cursor up
```

```
EC2B FE12               CPI   :12
EC2D DAABED             JC    :EDAB     Cursor down
EC30 CAD3ED             JZ    :EDD3     Cursor left
EC33 FE14               CPI   :14
EC35 DAF6ED             JC    :EDF6     Cursor right
EC38 CA4BEC             JZ    :EC4B     Window up
EC3B FE16               CPI   :16
EC3D DAB3EC             JC    :ECB3     Window down
EC40 CA50ED             JZ    :ED50     Window left
EC43 FE17               CPI   :17
EC45 CAF8EC             JZ    :ECF8     Window right
EC48 C34BEF             JMP   :EF4B     Insert character
                        *
                        *************
                        * WINDOW UP *
                        *************
                        *
                        * Moves window up one line.
                        *
                        * Exit: All registers preserved.
                        *       CY=0: Window unchanged.
                        *       CY=1: Window changed.
                        *
EC4B D5         EWUP    PUSH  D
EC4C E5                 PUSH  H
EC4D F5                 PUSH  PSW
EC4E 2AA900             LHLD  :00A9     Get offset of top of window
EC51 7C                 MOV   A,H
EC52 B5                 ORA   L
EC53 CADAEC             JZ    :ECDA     Window unchanged if
                                        offset = 0; CY=0
EC56 AF                 XRA   A
EC57 32AF00             STA   :00AF     Clear cursor pos in buffer
EC5A 3AAC00             LDA   :00AC     Get Y-offset cursor in
                                        in document
EC5D 95                 SUB   L         Minus offset top of window
EC5E FE17               CPI   :17       Nr of lines in window -1
EC60 CC88ED             CZ    :ED88     Cursor up if at
                                        bottom of window
EC63 2A7200             LHLD  :0072     Get cursor pos addr
EC66 E5                 PUSH  H
EC67 CD6BE3             CALL  :E36B     Delete cursor
EC6A CD74EC             CALL  :EC74     Move window, correct pntrs
EC6D E1                 POP   H
EC6E 117AFF     L2E172  LXI   D,:FF7A   Length one line
EC71 C3D5EC             JMP   :ECD5     Put cursor on next line,
                                        quit with CY=1
                        *
                        * SCROLL EDIT DISPLAY DOWN ONE LINE:
                        *
                        * Exit: BC preserved. AFDEHL corrupted.
                        *
EC74 C5         L2E173  PUSH  B
EC75 018600             LXI   B,:0086   Length one screen line
EC78 CD86EC             CALL  :EC86     Move window up in text
                                        1 line
EC7B D5                 PUSH  D
EC7C 2AA900             LHLD  :00A9     Get offset of top of window
EC7F 2B                 DCX   H         -1
EC80 22A900             SHLD  :00A9     And preserve it
EC83 C3EFEC             JMP   :ECEF     Print new line of text at
                                        window bottom
```

```
                  *
                  ***********************
                  * MOVE WINDOW IN TEXT *
                  ***********************
                  *
                  * Moves window up or left.
                  *
                  * Entry: BC: Offset (#86 for 1 line, #02 for
                  *              1 position).
                  * Exit:  All registers corrupted.
                  *
EC86 2ABC00       L2E174  LHLD   :008C      Get addr after end char area
EC89 54                   MOV    D,H        ) in DE
EC8A 5D                   MOV    E,L        )
EC8B 09                   DAD    B          Add offset
EC8C EB                   XCHG              HL: end char area;
                                            DE: end char area + offset
                  * Move full screen:

EC8D 010600       L2E175  LXI    B,:0006    3 not used locations
                                            at line end
EC90 09                   DAD    B          )
EC91 EB                   XCHG              ) Add it to both HL and DE
EC92 09                   DAD    B          ) and exchange DE and HL
EC93 EB                   XCHG              )
EC94 063C                 MVI    B,:3C      60 char/line visible

                  * Move one line:

EC96 23           L2E176  INX    H
EC97 23                   INX    H          New destination pntr
EC98 13                   INX    D
EC99 13                   INX    D          New origin pntr
EC9A 1A                   LDAX   D          Get one char
EC9B 77                   MOV    M,A        Move it to new screen loc
EC9C 05                   DCR    B          char count -1
EC9D C296EC               JNZ    :EC96      Next char if not ready
                  *
ECA0 010800               LXI    B,:0008    4 not-useable pos at line
                                            boundary
ECA3 09                   DAD    B          HL pnts to end previous line
                                            (destination)
ECA4 E5                   PUSH   H
ECA5 2A8A00               LHLD   :008A      Get addr start char area
ECA8 EB                   XCHG              in DE
ECA9 09                   DAD    B          Update origin too
ECAA CDFBE6               CALL   :E6FB      Compare HL-DE
ECAD EB                   XCHG              New origin in DE
ECAE E1                   POP    H          Destination in HL
ECAF DA8DEC               JC     :EC8D      Evt another line
ECB2 C9                   RET
                  *
                  ***************
                  * WINDOW DOWN *
                  ***************
                  *
                  * Moves window down one line.
                  *
                  * Exit: ABCDEHL preserved.
                  *       CY=0: Window unchanged.
                  *       CY=1: Window changed.
                  *
```

```
ECB3 D5         EWDN    PUSH  D
ECB4 E5                 PUSH  H
ECB5 F5                 PUSH  PSW
ECB6 2AAC00             LHLD  :00AC     Get Y-offset cursor in
                                        document
ECB9 3AA900             LDA   :00A9     Get offset of top of window
ECBC BD                 CMP   L         Cursor at top of screen ?
ECBD CCABED             CZ    :EDAB     Then cursor down
ECC0 D2DAEC             JNC   :ECDA     Quit with CY=0 if
                                        window unchanged
ECC3 AF                 XRA   A
ECC4 32AF00             STA   :00AF     Clear cursor pos in buffer
ECC7 2A7200             LHLD  :0072     Get cursor pos addr
ECCA E5                 PUSH  H
ECCB CD6BE3             CALL  :E36B     Delete cursor
ECCE CDDFEC             CALL  :ECDF     Scroll edit display up
                                        one line
ECD1 E1                 POP   H
ECD2 118600     L2E177  LXI   D,:0086   Length one screen line
ECD5 19         L2E178  DAD   D         One line further
ECD6 CD30E3     L2E179  CALL  :E330     Put new cursor on screen
ECD9 37                 STC             Exit CY=1 if cursor moved
ECDA E1         L2E180  POP   H
ECDB 7C                 MOV   A,H
ECDC E1                 POP   H
ECDD D1                 POP   D
ECDE C9                 RET
                *
                * SCROLL EDIT DISPLAY UP ONE LINE:
                *
                * Exit: BC preserved, AFDEHL corrupted.
                *
ECDF C5         L2E181  PUSH  B
ECE0 CDCBE1             CALL  :E1CB     Scroll window up 1 line
ECE3 E5                 PUSH  H
ECE4 2AA900             LHLD  :00A9     Get offset of top of window
ECE7 23                 INX   H         +1
ECE8 22A900             SHLD  :00A9     Preserve it
ECEB 111700             LXI   D,:0017   Nr of lines for window
ECEE 19                 DAD   D         HL pnts to new line at
                                        window bottom
ECEF CD1CEE     L2E182  CALL  :EE1C     Skip lines
ECF2 D1                 POP   D
ECF3 CDC0EE             CALL  :EEC0     Print new line of text
ECF6 C1                 POP   B
ECF7 C9                 RET
                *
                ***************
                * WINDOW RIGHT *
                ***************
                *
                * Moves window right one position.
                *
                * Exit: ABCDEHL preserved.
                *       CY=0: Window unchanged.
                *       CY=1: Window changed.
                *
ECF8 D5         EWRT    PUSH  D
ECF9 E5                 PUSH  H
ECFA F5                 PUSH  PSW
ECFB 3AA800             LDA   :00A8     Get offset of left side
                                        of window
```

```
ECFE FEC4                CPI   :C4        256-60 ?
ED00 CADAEC              JZ    :ECDA      Abort if window at right en
                                          of text
ED03 6F                  MOV   L,A        Offset in L
ED04 3AAB00              LDA   :00AB      Get X-offset cursor in
                                          document
ED07 BD                  CMP   L          Cursor at left side of
                                          window ?
ED08 CCF6ED              CZ    :EDF6      Then move cursor right
ED0B 2A7200              LHLD  :0072      Get cursor pos addr
ED0E E5                  PUSH  H
ED0F CD6BE3              CALL  :E36B      Delete cursor
ED12 CD1BED              CALL  :ED1B      Scroll display left 1 pos
ED15 E1                  POP   H
ED16 23          L2E184  INX   H
ED17 23                  INX   H
ED18 C3D6EC              JMP   :ECD6      Put cursor on screen;
                                          quit with CY=1
                 *
                 * SCROLL EDIT DISPAY LEFT ONE POSITION:
                 *
                 * Exit: BC preserved, AFDEHL corrupted.
                 *
ED1B C5          L2E185  PUSH  B
ED1C 01FEFF              LXI   B,:FFFE    -2
ED1F CDCEE1              CALL  :E1CE      Scroll window
ED22 3AA800              LDA   :00A8      Get offset of left side
                                          of window
ED25 3C                  INR   A          +1
ED26 32A800              STA   :00A8      Preserve it
ED29 C63B                ADI   :3B        +60 (offset right side
                                          of window)
ED2B 1182FF              LXI   D,:FF82    -7E
ED2E 47          L2E186  MOV   B,A        Offset right side of
                                          window in B
ED2F 0E18                MVI   C,:18      Nr of lines in window
ED31 2A8A00              LHLD  :008A      Get pntr to start char area
ED34 19                  DAD   D          Pos 60th char on screen
ED35 EB                  XCHG             in DE
ED36 2AA900              LHLD  :00A9      Get offset of top of window
ED39 CD1CEE              CALL  :EE1C      Skip lines

                 * Put newly visible 60th character on screen:

ED3C CD7BEE      L2E187  CALL  :EE7B      Skip to Bth pos on line
ED3F 12                  STAX  D          Next visible char in window
ED40 CD38EE              CALL  :EE38      Next line
ED43 E5                  PUSH  H
ED44 217AFF              LXI   H,:FF7A    -86 (length one line)
ED47 19                  DAD   D          Pnts to 60th char on
                                          next line
ED48 EB                  XCHG
ED49 E1                  POP   H
ED4A 0D                  DCR   C          Update line count
ED4B C23CED      L2E188  JNZ   :ED3C      Scroll next line if not
                                          ready
ED4E C1                  POP   B
ED4F C9                  RET
                 *
                 ***************
                 * WINDOW LEFT *
                 ***************
```

```
                        *
                        * Moves window left one position.
                        *
                        * Exit: ABCDEHL preserved.
                        *       CY=0: Window unchanged.
                        *       CY=1: Window changed.
                        *
ED50 D5                 EWLF    PUSH   D
ED51 E5                         PUSH   H
ED52 F5                         PUSH   PSW
ED53 3AA800                     LDA    :00A8     Get offset of left side
                                                 of window
ED56 B7                         ORA    A
ED57 CADAEC                     JZ     :ECDA     Abort if offset = 0
ED5A 6F                         MOV    L,A       Offset in L
ED5B 3AAB00                     LDA    :00AB     Get X-offset of cursor
                                                 in document
ED5E 95                         SUB    L
ED5F FE3B                       CPI    :3B       Cursor at right side of
                                                 window ?
ED61 CCD3ED                     CZ     :EDD3     Then cursor left
ED64 2A7200                     LHLD   :0072     Get cursor pos addr
ED67 E5                         PUSH   H
ED68 CD6BE3                     CALL   :E36B     Delete cursor
ED6B CD74ED                     CALL   :ED74     Move window, correct pntrs
ED6E E1                         POP    H
ED6F 2B                 L2E189  DCX    H
ED70 2B                         DCX    H         New cursor pos
ED71 C3D6EC                     JMP    :ECD6     Put cursor on screen;
                                                 quit with CY=1
                        *
                        * SCROLL EDIT DISPLAY RIGHT ONE POSITION:
                        *
                        * Exit: BC preserved; AFDEHL corrupted.
                        *
ED74 C5                 L2E190  PUSH   B
ED75 010200                     LXI    B,:0002
ED78 CD86EC                     CALL   :EC86     Move window in text
ED7B 3AA800                     LDA    :00AB     Get offset of left side
                                                 of window
ED7E 3D                         DCR    A         -1
ED7F 32A800                     STA    :00A8     Preserve it
ED82 11F8FF                     LXI    D,:FFF8   Gives in ED2E 1st pos
                                                 on 1st screen line
ED85 C32EED                     JMP    :ED2E     Scroll display left
                        *
                        *************
                        * CURSOR UP *
                        *************
                        *
                        * Moves cursor up one position.
                        *
                        * Exit: ABCDEHL preserved.
                        *       CY=0: Cursor not moved.
                        *       CY=1: Cursor moved.
                        *
ED88 D5                 ECUP    PUSH   D
ED89 E5                         PUSH   H
ED8A F5                         PUSH   PSW
ED8B 2AAC00                     LHLD   :00AC     Get Y-offset cursor in
                                                 document
ED8E 7C                         MOV    A,H
```

```
ED8F B5                       ORA   L
ED90 CADAEC                   JZ    :ECDA     Quit if cursor at top
                                              left; CY=0
ED93 AF                       XRA   A
ED94 32AF00                   STA   :00AF     Clear cursor pos in buffer
ED97 3AA900                   LDA   :00A9     Get offset of top of window
ED9A BD                       CMP   L         Cursor at top of window ?
ED9B CC4BEC                   CZ    :EC4B     Then window up
ED9E 2B                       DCX   H
ED9F 22AC00                   SHLD  :00AC     Store Y-offset cursor in
                                              document
EDA2 2A7200                   LHLD  :0072     Get cursor pos addr
EDA5 CD6BE3                   CALL  :E36B     Delete cursor
EDA8 C3D2EC                   JMP   :ECD2     Put cursor on new pos;
                                              quit with CY=1
                    *
                    ***************
                    * CURSOR DOWN *
                    ***************
                    *
                    * Moves cursor down one position.
                    *
                    * Exit: ABCDEHL preserved.
                    *         CY=0: Cursor not moved.
                    *         CY=1: Cursor moved.
                    *
EDAB D5             ECDN      PUSH  D
EDAC E5                       PUSH  H
EDAD F5                       PUSH  PSW
EDAE 2AAC00                   LHLD  :00AC     Get Y-offset cursor in
                                              in document
EDB1 23                       INX   H         +1
EDB2 E5                       PUSH  H         Preserve it
EDB3 CD1CEE                   CALL  :EE1C     Skip lines
EDB6 E1                       POP   H
EDB7 D2DAEC                   JNC   :ECDA     No move if cursor at last
                                              line of text; CY=0
EDBA AF                       XRA   A
EDBB 32AF00                   STA   :00AF     Clear cursor pos in buffer
EDBE 3AA900                   LDA   :00A9     Get offset of top of window
EDC1 C618                     ADI   :18       24 lines in window
EDC3 BD                       CMP   L         Cursor at window bottom ?
EDC4 CCB3EC                   CZ    :ECB3     Then window down
EDC7 22AC00                   SHLD  :00AC     Store Y-offset cursor in
                                              document
EDCA 2A7200                   LHLD  :0072     Get cursor pos addr
EDCD CD6BE3                   CALL  :E36B     Delete cursor
EDD0 C36EEC                   JMP   :EC6E     Put cursor on new pos;
                                              quit with CY=1
                    *
                    ***************
                    * CURSOR LEFT *
                    ***************
                    *
                    * Moves cursor left one position.
                    *
                    * Exit: ABCDEHL preserved.
                    *         CY=0: Cursor not moved.
                    *         CY=1: Cursor moved.
                    *
EDD3 D5             ECLF      PUSH  D
EDD4 E5                       PUSH  H
```

```
EDD5 F5                        PUSH  PSW
EDD6 3AA800                    LDA   :00A8     Get offset of left side
                                               of window
EDD9 6F                        MOV   L,A       in L
EDDA 3AAB00                    LDA   :00AB     Get X-offset cursor in
                                               document
EDDD B7                        ORA   A
EDDE CADAEC                    JZ    :ECDA     Abort if X-offset = 0
EDE1 BD                        CMP   L         Cursor at left side of
                                               window ?
EDE2 CC50ED                    CZ    :ED50     Then window left
EDE5 3D                        DCR   A         X-offset -1
EDE6 32AB00                    STA   :00AB     Preserve it
EDE9 AF                        XRA   A
EDEA 32AF00                    STA   :00AF     Clear cursor pos in buffer
EDED 2A7200                    LHLD  :0072     Get cursor pos addr
EDF0 CD6BE3                    CALL  :E36B     Delete cursor
EDF3 C316ED                    JMP   :ED16     Put cursor on new pos;
                                               quit with CY=1
                      *
                      *
                      *
EDF6                           END

**************************
* S Y M B O L   T A B L E *
**************************

ECDN    EDAB   ECLF    EDD3   ECUP    ED88   EINIT   EBF4
EOBEY   EC1E   EWDN    ECB3   EWLF    ED50   EWRT    ECF8
EWUP    EC4B   L2E172  EC6E   L2E173  EC74   L2E174  EC86
L2E175  EC8D   L2E176  EC96   L2E177  ECD2   L2E178  ECD5
L2E179  ECD6   L2E180  ECDA   L2E181  ECDF   L2E182  ECEF
L2E184  ED16   L2E185  ED1B   L2E186  ED2E   L2E187  ED3C
L2E188  ED4B   L2E189  ED6F   L2E190  ED74
```

```
                         ORG   :EDF6
                *
                *
                *
                ****************
                * CURSOR RIGHT *
                ****************
                *
                * Moves cursor right one position.
                *
                * Exit: ABCDEHL preserved.
                *       CY=0: Cursor not moved.
                *       CY=1: Cursor moved.
                *
EDF6 D5         ECRT     PUSH  D
EDF7 E5                  PUSH  H
EDF8 F5                  PUSH  PSW
EDF9 3AA800              LDA   :00A8     Get offset of left side
                                         of window
EDFC C63B                ADI   :3B       Calc end of window
EDFE 6F                  MOV   L,A       Result in L
EDFF 3AAB00              LDA   :00AB     Get X-offset of cursor
                                         in document
EE02 FEFF                CPI   :FF       Max nr char/line reached ?
EE04 CADAEC              JZ    :ECDA     Then quit with CY=0
EE07 BD                  CMP   L         End of window reached ?
EE08 CCF8EC              CZ    :ECF8     Then window right
EE0B 3C                  INR   A         Incr cursor pos on line
EE0C 32AB00              STA   :00AB     Store X-offset of cursor
                                         in document
EE0F AF                  XRA   A
EE10 32AF00              STA   :00AF     Clear cursor pos in buffer
EE13 2A7200              LHLD  :0072     Get cursor pos addr
EE16 CD6BE3              CALL  :E36B     Delete old cursor
EE19 C36FED              JMP   :ED6F     Put new cursor on screen;
                                         quit with CY=1
                *
                ****************************
                * SKIP LINES IN EDIT BUFFER *
                ****************************
                *
                * Entry: HL: Number of lines to be skipped.
                * Exit:  HL: Points to 1st not skipped line.
                *        CY=1: O.K.
                *        CY=0: End of text reached before count
                *              is zero.
                *        ABCDE preserved.
                *
EE1C D5         L2E191   PUSH  D
EE1D F5                  PUSH  PSW
EE1E EB                  XCHG            Count in DE
EE1F 2AA200              LHLD  :00A2     Get startaddr editbuf
EE22 7A         L2E192   MOV   A,D       )
EE23 B3                  ORA   E         ) Abort with CY=1 if ready
EE24 CA33EE              JZ    :EE33     )
EE27 CD38EE              CALL  :EE38     Next line
EE2A 00                  NOP
EE2B B7                  ORA   A
EE2C CA34EE              JZ    :EE34     Abort with CY=0 if end of
                                         buffer reached before DE=0
EE2F 1B                  DCX   D         Count -1
EE30 C322EE              JMP   :EE22     Skip next line
```

```
EE33 37          L2E193  STC
EE34 D1          L2E194  POP    D
EE35 7A                  MOV    A,D
EE36 D1                  POP    D
EE37 C9                  RET
                 *
                 *************
                 * NEXT LINE *
                 *************
                 *
                 * Entry: HL: points to text, each line ended by
                 *            a car.ret, last line followed by 0.
                 * Exit:  HL: Points to next line.
                 *         A:  0 if end of text reached, else CR.
                 *         BCDE preserved.
                 *
EE38 7E          L2E195  MOV    A,M         Get char from text
EE39 B7                  ORA    A
EE3A CA43EE              JZ     :EE43       Ready if end of text reached
EE3D 23                  INX    H
EE3E FE0D                CPI    :0D         Car.ret ?
EE40 C238EE              JNZ    :EE38       Loop till end of line found
EE43 C9          L2E196  RET
                 *
                 ****************************************
                 * FIND CURRENT POSITION IN EDIT BUFFER *
                 ****************************************
                 *
                 * Exit: CY=1: HL: (00AF/AE) = cursor position in
                 *                 buffer.
                 *       CY=0: HL: Points to cursor line in the
                 *                 edit buffer.
                 *       ABCDE preserved.
                 *
EE44 2AAE00      L2E197  LHLD   :00AE       Get pntr to cursor pos
                                            in buffer
EE47 25                  DCR    H
EE48 24                  INR    H
EE49 37                  STC
EE4A C0                  RNZ                Quit (CY=1) if (AF) <> 0
EE4B D5                  PUSH   D
EE4C C5                  PUSH   B
EE4D F5                  PUSH   PSW
EE4E 2AAC00              LHLD   :00AC       Get Y-offset of cursor
                                            in document (no lines to
                                            be skipped)
EE51 54                  MOV    D,H         ) in DE
EE52 5D                  MOV    E,L         )
EE53 CD1CEE              CALL   :EE1C       Skip lines
EE56 22B200              SHLD   :00B2       Store pntr to start cursor
                                            line in buffer
EE59 E5                  PUSH   H           and preserve it
EE5A 2AA900              LHLD   :00A9       Get offset of top of window
EE5D CDF2E6              CALL   :E6F2       Calc difference
EE60 3E86                MVI    A,:86       Length one line
EE62 CD46EB              CALL   :EB46       Calc total length
EE65 EB                  XCHG               Result in DE
EE66 2A8A00              LHLD   :008A       Get startaddr char area
EE69 19                  DAD    D           Add offset
EE6A 22B000              SHLD   :00B0       Preserve pntr to start
                                            cursor line on screen
EE6D E1                  POP    H           Get pntr to start cursor
```

```
                                                line in buffer
EE6E 3AAB00                LDA    :00AB         Get X-offset of cursor
                                                in document
EE71 47                    MOV    B,A           in B
EE72 CD7BEE                CALL   :EE7B         Skip to Bth pos on line
                                                exit: HL pnts to cursor pos
EE75 C1                    POP    B
EE76 78                    MOV    A,B
EE77 C1                    POP    B
EE78 D1                    POP    D
EE79 3F                    CMC                  Abort with CY=0
EE7A C9                    RET
                   *
                   ************************************
                   * SKIP TO Bth POSITION ON TEXT LINE *
                   ************************************
                   *
                   * Looks through a textline until Bth position is
                   * found. On exit, character on this position is
                   * in A.
                   *
                   * Entry: HL: Start text line.
                   *        B:  Number of position in line.
                   * Exit:  CY=0: Position found:
                   *              A:  Character on that position.
                   *              HL: Points to Bth position.
                   *              BCDE preserved.
                   *        CY=1: Bth position is beyond car.ret,
                   *              tab or end of text.
                   *              HL: Points to CR, tab or 0.
                   *              A:  Space.
                   *
EE7B C5            L2E198  PUSH   B
EE7C D5                    PUSH   D
EE7D 0E00                  MVI    C,:00         Init count
EE7F C3FDD1        L2E199  JMP    :D1FD         Evaluate character
EE82 CAA3EE        L2E200  JZ     :EEA3         Abort if Bth pos reached
EE85 B7                    ORA    A             Set flags on char
EE86 CAA0EE                JZ     :EEA0         Jump if end of text reached
EE89 FE0D                  CPI    :0D
EE8B CAA0EE                JZ     :EEA0         Jump if char is car.ret
EE8E FE09                  CPI    :09
EE90 CA98EE                JZ     :EE98         Jump if char is tab
EE93 0C                    INR    C             Incr count
EE94 23            L2E201  INX    H             Pnts to next pos on line
EE95 C37FEE                JMP    :EE7F         Loop untill ready

                   * If character is tab:

EE98 CDA6EE        L2E202  CALL   :EEA6         Tabulate
EE9B 78                    MOV    A,B           Reqd pos in A
EE9C B9                    CMP    C             Compare with tab stops
EE9D D294EE                JNC    :EE94         Continu if not past tab

                   * If end of line or end of text reached or if
                   * past tab:

EEA0 3E20          L2E203  MVI    A,:20         Set char is space
EEA2 37                    STC                  Abort with CY=1
EEA3 D1            L2E204  POP    D
EEA4 C1                    POP    B
EEA5 C9                    RET
```

```
                        *
                        ************
                        * TABULATE *
                        ************
                        *
                        * Routine goes through tab-table for 1st
                        * tab-stop > C.
                        *
                        * Entry: 00B4/B5:  Pointer to tab-table.
                        *        C:        Line position.
                        * Exit:  If found: Tab in C.
                        *        Else:     C = C + 1.
                        *        AFBDEHL preserved.
                        *
EEA6 F5                 L2E205  PUSH   PSW
EEA7 C5                         PUSH   B
EEA8 E5                         PUSH   H
EEA9 2AB400                     LHLD   :00B4      Get addr tab table
EEAC 41                 L2E206  MOV    B,C        Line pos in B
EEAD 04                         INR    B          +1
EEAE 7E                         MOV    A,M        Get tab from table
EEAF B7                         ORA    A
EEB0 CABAEE                     JZ     :EEBA      If end tab table reached
EEB3 23                         INX    H          Pnts to next tab
EEB4 47                         MOV    B,A        Tab in B
EEB5 79                         MOV    A,C        Line pos in A
EEB6 B8                         CMP    B
EEB7 D2ACEE                     JNC    :EEAC      Get next tab if line
                                                  pos > tab stop
EEBA 48                 L2E207  MOV    C,B        Replace line pos by tab stop
                                                  or by line pos +1 if no tab
                                                  found
EEBB E1                         POP    H
EEBC F1                         POP    PSW
EEBD 47                         MOV    B,A
EEBE F1                         POP    PSW
EEBF C9                         RET
                        *
                        ******************************
                        * PRINT LINE OF TEXT IN WINDOW /
                        ******************************
                        *
                        * Entry: HL:   Address of textline in buffer.
                        *        DE:   Line control byte of screen line
                        *              to print text on.
                        *        00A8: Number of non-printing positions.
                        *
                        * During execution loop:
                        *        B:   Nr of non-printing pos left +1.
                        *        C:   Nr of screen line pos left.
                        *        D:   Current text line position.
                        *        E:   Count of blanks inserted.
                        *        HL:  Points to text.
                        *        TOS: Points to screen.
                        *
EEC0 C5                 L2E208  PUSH   B
EEC1 EB                         XCHG
EEC2 01F8FF                     LXI    B,:FFF8
EEC5 09                         DAD    B          1st printable pos on screen
EEC6 EB                         XCHG              in DE
EEC7 3AA800                     LDA    :00A8      Get offset of left side
                                                  of window
```

```
EECA 3C                INR   A
EECB 47                MOV   B,A        B is offset +1
EECC 0E3C              MVI   C,:3C      60 visible characters
EECE D5                PUSH  D          Preserve start screen line
EECF 1600              MVI   D,:00      ) Init current text pos
EED1 5A                MOV   E,D        ) and blanks count

              * Loop:

EED2 3E20     L2E209   MVI   A,:20      Space
EED4 1D                DCR   E
EED5 1C                INR   E
EED6 C2ECEE            JNZ   :EEEC      E<>0: Update screen pos pntr
EED9 7E                MOV   A,M        Else: Get char from buffer
EEDA B7                ORA   A          Set flags on char
EEDB C3D1D1            JMP   :D1D1      Test character
EEDE FE0D     L2E210   CPI   :0D
EEE0 CAECEE            JZ    :EEEC      Skip tab-handling if CR
EEE3 1E00              MVI   E,:00      Reset blanks count
EEE5 23                INX   H          Pnts to next char in text
EEE6 FE09              CPI   :09        Tab ?
EEE8 CA08EF            JZ    :EF08      Then tab-handling
EEEB 1C       L2E211   INR   E
EEEC 1D       L2E212   DCR   E          Update blanks count
EEED 14                INR   D          Update line position
EEEE 05                DCR   B          and nr of pos left in window
EEEF C2D2EE            JNZ   :EED2      No printing, cont with
                                        next char
EEF2 04                INR   B
EEF3 E3                XTHL             Screen pos in HL
EEF4 77                MOV   M,A        Display char on screen
EEF5 2B                DCX   H
EEF6 2B                DCX   H          Next screen pos
EEF7 E3                XTHL             Preserve it
EEF8 0D                DCR   C          End of screen line reached ?
EEF9 C2D2EE            JNZ   :EED2      Next char if not
EEFC CD38EE            CALL  :EE38      Else: next line
EEFF E3                XTHL             HL is screen pos
EF00 11FAFF            LXI   D,:FFFA
EF03 19                DAD   D          HL is 1st useable pos
                                        on next line
EF04 EB                XCHG             into DE
EF05 E1                POP   H
EF06 C1                POP   B
EF07 C9                RET

              * Tab-handling:

EF08 C5       L2E213   PUSH  B
EF09 4A                MOV   C,D        C is pos on current line
EF0A CDA6EE            CALL  :EEA6      Tabulate
EF0D 79                MOV   A,C        Next tab stop in A
EF0E 92                SUB   D          ) Calc blanks to be inserted
EF0F 3D                DCR   A          )
EF10 5F                MOV   E,A        Update blanks count
EF11 C1                POP   B
EF12 3E09              MVI   A,:09      Restore tab char in A
EF14 C3EBEE            JMP   :EEEB
              *
              ***************************
              * PRINT A COMPLETE WINDOW *
              ***************************
```

```
                  *
                  * Text is printed in window from (DE) to bottom
                  * text screen if text ends. A ASCII 0 (vertical
                  * bar) is printed at the beginning of all
                  * following lines.
                  *
                  * Entry: HL: Points to text.
                  *        DE: Points to line control byte screen line
                  * Exit:  HL: Points to next line to print.
                  *        DE: Bottom text screen.
                  *        AF corrupted, BC preserved.
                  *
EF17 E5           L2E214  PUSH   H
EF18 2A8C00               LHLD   :008C    Get bottom text screen
EF1B CDFBE6               CALL   :E6FB    Compare HL-DE
EF1E E1                   POP    H
EF1F CA28EF               JZ     :EF28    Quit if window full
EF22 CDC0EE               CALL   :EEC0    Print a textline
EF25 C317EF               JMP    :EF17    Loop until ready
EF28 C9           L2E215  RET
                  *
                  *************************************
                  * PRINT A TEXT LINE POINTED BY (00B2) *
                  * ON SCREEN POSITION (00B0) IN WINDOW *
                  *************************************
                  *
                  * Exit: All registers preserved.
                  *
EF29 F5           L2E216  PUSH   PSW
EF2A C5                   PUSH   B
EF2B D5                   PUSH   D
EF2C E5                   PUSH   H
EF2D 2AB000               LHLD   :00B0    Get pntr to start cursor
                                          line on screen
EF30 EB                   XCHG            in DE
EF31 2AB200               LHLD   :00B2    Get pntr to start cursor
                                          line in buffer
EF34 CDC0EE               CALL   :EEC0    Print a text line
EF37 C338E1               JMP    :E138    Popall, ret
                  *
                  *********************
                  * PRINT FULL SCREEN *
                  *********************
                  *
                  * Prints from Nth line in full screen window.
                  * N is the offset of the top of the window from
                  * the start of the edit buffer.
                  *
                  * Exit: All registers preserved.
                  *
EF3A F5           L2E217  PUSH   PSW
EF3B C5                   PUSH   B
EF3C D5                   PUSH   D
EF3D E5                   PUSH   H
EF3E 2A8A00               LHLD   :008A    Get startaddr char area
EF41 EB                   XCHG            in DE
EF42 2AA900               LHLD   :00A9    Get offset of top of window
EF45 CD1CEE               CALL   :EE1C    Skip lines
EF48 C385CE               JMP    :CE85    Print complete window
                  *
```

```
                ******************************
                * INSERT CHARACTER IN BUFFER *
                ******************************
                *
                * Entry: Character in A.
                * Exit:  ABCDEHL preserved.
                *        CY=0: Buffer full, no action.
                *        CY=1: Character inserted.
                *
EF4B C5         EINCH   PUSH   B
EF4C F5                 PUSH   PSW
EF4D D5                 PUSH   D
EF4E E5                 PUSH   H
EF4F 47                 MOV    B,A      Char in B
EF50 2AA600             LHLD   :00A6    Get end available space
EF53 EB                 XCHG            in DE
EF54 2AA400             LHLD   :00A4    Get input pointer
EF57 CDFBE6             CALL   :E6FB    Compare HL-DE
EF5A EB                 XCHG            Input pntr in E
EF5B 78                 MOV    A,B      Char in A
EF5C DC44EE             CC     :EE44    Space available: Find
                                        current pos in buffer
EF5F D296EF             JNC    :EF96    Buffer full: quit, char in A
EF62 E5                 PUSH   H        Preserve end available space
EF63 2A7200             LHLD   :0072    Get cursor pos addr
EF66 E3                 XTHL            Put it on stack
EF67 E5                 PUSH   H
EF68 CD6BE3             CALL   :E36B    Delete cursor
EF6B EB                 XCHG            Old input pntr in HL
EF6C 00                 NOP
EF6D 23                 INX    H        +1
EF6E 22A400             SHLD   :00A4    Update input pointer
EF71 2B                 DCX    H
EF72 44                 MOV    B,H      ) Old input pntr in BC
EF73 4D                 MOV    C,L      )
EF74 2B                 DCX    H
EF75 EB                 XCHG            Old input pntr -1 in DE
EF76 E1                 POP    H        Get end available space
EF77 2B                 DCX    H        -1
EF78 CDC2E6             CALL   :E6C2    Move screen area 1 pos
EF7B 23                 INX    H
EF7C 77                 MOV    M,A      Insert char in buffer
EF7D 23                 INX    H
EF7E 22AE00             SHLD   :00AE    Preserve pntr to cursor
                                        in buffer
EF81 E1                 POP    H        Get end available space
EF82 FE0D               CPI    :0D
EF84 CA9CEF             JZ     :EF9C    Jump if char is car.ret
EF87 CD29EF             CALL   :EF29    Reprint text line in window
EF8A CD30E3             CALL   :E330    Put cursor on screen
EF8D FE09               CPI    :09
EF8F CAB9EF             JZ     :EFB9    Jump if char is tab
EF92 CDF6ED             CALL   :EDF6    Move cursor right 1 pos
EF95 37         L2E218  STC
EF96 E1         L2E219  POP    H
EF97 D1                 POP    D
EF98 C1                 POP    B
EF99 78                 MOV    A,B
EF9A C1                 POP    B
EF9B C9                 RET

                * If car.ret:
```

```

EF9C AF           L2E220  XRA    A
EF9D 32A800               STA    :00A8      )
EFA0 32AB00               STA    :00AB      ) Reset pointers
EFA3 32AF00               STA    :00AF      )
EFA6 CD3AEF               CALL   :EF3A      Reprint full screen
EFA9 2AB000               LHLD   :00B0      Get pntr to start cursor
                                            line on screen
EFAC 11F8FF               LXI    D,:FFF8
EFAF 19                   DAD    D          New cursor addr
EFB0 CD30E3               CALL   :E330      Put cursor on screen
EFB3 CDABED               CALL   :EDAB      Move cursor down
EFB6 C395EF               JMP    :EF95      Quit with CY=1

                  * If tab:

EFB9 CDF6ED       L2E230  CALL   :EDF6      Move cursor right
EFBC 3AAB00               LDA    :00AB      Get X-offset of cursor
                                            in document
EFBF 47                   MOV    B,A        in B
EFC0 2AB200               LHLD   :00B2      Get pntr to start cursor
                                            line in buffer
EFC3 CD7BEE               CALL   :EE7B      Skip to Bth pos on line
EFC6 DAB9EF               JC     :EFB9      Again
EFC9 C395EF               JMP    :EF95      Abort with CY=1
                  *
                  ******************************
                  * DELETE CHARACTER IN BUFFER *
                  ******************************
                  *
                  * Deletes character at cursor position and moves
                  * the text above this position. No action if past
                  * car.ret, tab or at or beyond end of text.
                  *
                  * Exit: ABCDEHL preserved.
                  *       CY=1: O.K.
                  *       CY=0: If past car.ret, tab or beyond 0.
                  *
EFCC C5           EDLCH   PUSH   B
EFCD F5                   PUSH   PSW
EFCE D5                   PUSH   D
EFCF E5                   PUSH   H
EFD0 CD44EE               CALL   :EE44      Find current pos in buffer
EFD3 D296EF               JNC    :EF96      Quit if past CR, tab
                                            or 0 with CY=0
EFD6 7E                   MOV    A,M        Get char
EFD7 B7                   ORA    A
EFD8 CA96EF               JZ     :EF96      If at end of text:
                                            quit with CY=0
EFDB 00                   NOP
EFDC 00                   NOP
EFDD 00                   NOP
EFDE E5                   PUSH   H
EFDF 2A7200               LHLD   :0072      Get cursor pos addr
EFE2 E3                   XTHL              preserve it on stack
EFE3 CD6BE3               CALL   :E36B      Delete cursor
EFE6 EB                   XCHG              Original HL in DE
EFE7 2AA400               LHLD   :00A4      Get input pointer
EFEA 2B                   DCX    H          Decrement it
EFEB 22A400               SHLD   :00A4      And store update input pntr
EFEE 44                   MOV    B,H        ) Move updated pointer
EFEF 4D                   MOV    C,L        ) into BC
```

```
498 EFF0 0B                        DCX   B        Decrement it once again
499 EFF1 EB                        XCHG           restore original HL
500 EFF2 CDC2E6                    CALL  :E6C2    Move screen area above
501                                               downwards
502 EFF5 FE0D                      CPI   :0D      Deleted car.ret ?
503 EFF7 C429EF                    CNZ   :EF29    If not: reprint text line
504 EFFA CC3AEF                    CZ    :EF3A    Else: reprint full screen
505 EFFD C3F2CE                    JMP   :CEF2    Put cursor on screen, abort
506                                               with CY=1
507                    *
508                    *
509                    *
510 F000                           END

***************************
* S Y M B O L   T A B L E *
***************************

ECRT    EDF6    EDLCH   EFCC    EINCH   EF4B    L2E191  EE1C
L2E192  EE22    L2E193  EE33    L2E194  EE34    L2E195  EE38
L2E196  EE43    L2E197  EE44    L2E198  EE7B    L2E199  EE7F
L2E200  EE82    L2E201  EE94    L2E202  EE98    L2E203  EEA0
L2E204  EEA3    L2E205  EEA6    L2E206  EEAC    L2E207  EEBA
L2E208  EEC0    L2E209  EED2    L2E210  EEDE    L2E211  EEEB
L2E212  EEEC    L2E213  EF08    L2E214  EF17    L2E215  EF28
L2E216  EF29    L2E217  EF3A    L2E218  EF95    L2E219  EF96
L2E220  EF9C    L2E230  EFB9
```