QS-Forth Screens#

```
SCR # 1
  0    FORTH DEFINITIONS HEX
  1
  2  : L-ON  0780 ' CREATE ! ;
  3  : L-OFF ' FIRST CFA
  4          ' CREATE ! ; DECIMAL
  5  : LOCATE ~[COMPILE] ' DUP
  6         ~[ FENCE @ ] LITERAL >
  7    IF NFA 2 - @ DUP 0=
  8    IF    25 MESSAGE
  9    ELSE   DUP 0 <
 10    IF 5 MESSAGE QUIT ELSE LIST
 11    ENDIF ENDIF
 12    ELSE  9 MESSAGE ENDIF ;
 13
 14  DECIMAL -->
 15

SCR # 2
  0  HEX
  1
  2  : CASE: <BUILDS ] SMUDGE
  3          DOES> SWAP 2 * +
  4          @ EXECUTE ;
  5
  6  6 USER S0  ( COMP STK ORG )
  7  8 USER R0  ( RET STK ORG )
  8
  9 : UPDATE PREV @ @ 8000 OR PREV
 10     @ ! ;
 11
 12 : DCX DECIMAL ;
 13
 14   DECIMAL -->
 15

SCR # 3
  0  ( STACK WORDS )
  1
  2  : .S ( PEEK AT STACK )
  3        S0 @  SP@ -  2 / 1 -
  4        IF SP@ 2 - S0 @  2 -
  5           DO I @ . -2 +LOOP
  6        ELSE ." STACK EMPTY "
  7        CR ENDIF ;
  8
  9  : 2DUP OVER OVER ;
 10
 11  : LOAD-ED     30 LOAD ;
 12  : LOAD-ASM    58 LOAD ;
 13  : LOAD-IO     6 LOAD ;
 14          DECIMAL -->
 15

SCR # 4
  0  ( TEXT LINE  )        HEX
```

```
 1
 2 : TEXT ( TEXT --> PAD )
 3     HERE C/L 1+ BLANKS WORD HERE
 4     PAD C/L 1+ CMOVE ;
 5
 6 : LINE  ( ADR OF LINE-->STK )
 7     DUP FFF0 AND 17 ?ERROR
 8     SCR @ (LINE) DROP ;
 9   : INVON FF 0668 C!
10           FF 06BE C! ;
11   : INVOF 7F 0668 C!
12           7F 06BE C! ;
13   : BELL  06BE C@ FF 06BE C!
14       FD EMIT 06BE C!  ;
15   INVOF   DECIMAL -->

SCR # 5
 0  ( HDUMPER )   HEX
 1   : HXOT <# # # #> TYPE  ;
 2   : HPRT     C@ 7F AND DUP
 3        20 < IF DROP 2E THEN
 4  SPACE 1B EMIT EMIT SPACE ;
 5   : ADDR <# # # # # #> TYPE ;
 6   : DUMP ( ADDR CNT ) CR
 7       HEX 1 - 08 / 1+
 8        0 DO DUP 0 ADDR SPACE
 9      8 0 DO DUP I + C@ 0
10        HXOT  SPACE  LOOP CR
11           4 SPACES
12    8 0 DO DUP I + HPRT LOOP
13      08 + CR    LOOP DROP ;
14   : U. 0 <# #S #> TYPE SPACE  ;
15    DECIMAL    ;S

SCR # 6
 0   FORTH DEFINITIONS
 1  ( CIO CALL CHEATER )  HEX
 2   ." I/O MODULE LOADING..." CR
 3   CREATE JSRCIO ( CALL TO )
 4                 ( CH #6   )
 5     B5 C, 00 C, ( LDA TOS )
 6     86 C, B5 C, ( STX XSAVE )
 7     A2 C, 60 C, ( LDX #$60  )
 8     20 C, C4 C, E4 C, ( JSR )
 9     A6 C, B5 C, ( LDX XSAVE )
10     E8 C, E8 C, ( CLR STK  )
11     4C C, DF C, 0A C, ( 0A  )
12
13     SMUDGE
14
15           DECIMAL -->

SCR # 7
 0  ( IOCB CONTSTANTS  )    HEX
 1
 2  60 VARIABLE IO#
 3
 4  : IO#@ IO# @ + ;
 5
```

```
  6  : CHANID 340 IO#@   ;
  7  : CDEV#  341 IO#@   ;
  8  : CCMD   342 IO#@   ;
  9  : CSTAT  343 IO#@   ;
 10  : BUFADR 344 IO#@   ;
 11  : BUFLEN 348 IO#@   ;
 12  : CAUX1  34A IO#@   ;
 13  : CAUX2  34B IO#@   ;
 14
 15  DECIMAL -->

SCR # 8
  0  ( IOCB COMMANDS )       HEX
  1
  2  04 CONSTANT INOP
  3  08 CONSTANT OUTOP
  4
  5  : #-> ( ASSIGNS I0CB # )
  6     10 *  DUP IO# !
  7           ' JSRCIO 5 +  C! ;
  8
  9  0 VARIABLE "K" -2 ALLOT
 10     4B C, 3A C, 9B C,
 11
 12  0 VARIABLE "S" -2 ALLOT
 13     53 C, 3A C, 9B C,
 14
 15  DECIMAL -->

SCR # 9
  0  ( IOCB CONT )
  1  HEX
  2  0 VARIABLE "P" -2 ALLOT
  3     50 C, 3A C, 9B C,
  4
  5  0 VARIABLE "C" -2 ALLOT
  6     43 C, 3A C, 9B C,
  7
  8  0 VARIABLE "E" -2 ALLOT
  9     45 C, 3A C, 9B C,
 10
 11  : CKSTAT CSTAT C@ DUP 80 AND
 12      IF 7F AND 21 + DUP ?ERROR
 13         ELSE DROP ENDIF ;
 14                DECIMAL -->
 15

SCR # 10
  0  ( OPEN IOCB )
  1   HEX
  2  ( "K" INOP 0 I0CB# OPEN )
  3
  4  : OPEN #->        ( IOCB# )
  5         03 CCMD C! ( OPEN CMD)
  6         CAUX2 C!   ( 0 USUAL )
  7         CAUX1 C!   ( IN/OUT  )
  8         BUFADR !   ( -> K:   )
  9         JSRCIO     ( SET TO 6)
 10         CKSTAT ;   ( ERROR?  )
```

```
 11
 12
 13           DECIMAL -->
 14
 15

SCR # 11
  0  ( GET A CHAR TO STACK ) HEX
  1  ( GET ... ASCII TO STACK )
  2
  3  : GET #->     0 ( DUMMY )
  4        07 CCMD C! ( GET CHAR )
  5        0  BUFADR ! ( 0 -> A  )
  6        JSRCIO  CKSTAT ;
  7
  8  : PUT #->
  9        0B CCMD C! ( PUT CHAR )
 10        0 BUFADR ! ( 0-> A   )
 11        JSRCIO  CKSTAT DROP  ;
 12   HEX
 13  : CLOSE #->
 14          0C CCMD C! JSRCIO
 15          CKSTAT ; DECIMAL -->

SCR # 12
  0  ( PRINTER WORDS )
  1
  2  : PRON  4 CLOSE
  3      "P" OUTOP 0 4 OPEN ;
  4
  5  : PROF 4 CLOSE
  6      "E" OUTOP 0 4 OPEN ;
  7
  8          -->
  9
 10
 11
 12
 13
 14
 15

SCR # 13
  0  ( VOL,DIST,FREQ,VOICE, SND )
  1  HEX
  2   : SOUND 0232 C@ 07 AND
  3      D20F C!  0 D208 C!
  4      DUP 3 > IF
  5  ." ILLEGAL CHAN" ABORT ENDIF
  6     2 * D200 + >R >R 10 * OR
  7     EF AND
  8     100 * R>  OR R> ! ;
  9
 10
 11   : XSND D208 D200 DO
 12           0 I C! LOOP ;
 13   DECIMAL
 14   -->
 15
```

```
SCR # 14
  0  ." GRAPHICS LOADING..." CR
  1  ( ALL USE CH # 6 )
  2  : GRN 6 OPEN ;
  3  : GR.  ( MODE 7 SPLIT )
  4      6 CLOSE >R
  5      "S" OUTOP 16 OR R> GRN ;
  6
  7  : GR.16 6 CLOSE >R "S"
  8      OUTOP R> GRN  ;  HEX
  9
 10  : SETCOLOR  DUP 4 > IF
 11     ." ILLEGAL COLOR"
 12      . . .           ELSE
 13     02C4 + >R 10 * OR
 14       R>  C!         ENDIF ;
 15     DECIMAL -->

SCR # 15
  0  ( PLOT DRAWTO ) HEX
  1  : CKER CSTAT C@ 8D = IF
  2            ." RANGE ERROR "
  3    QUIT ELSE CKSTAT ENDIF ;
  4
  5  : DRAWTO ( Y,X,C ) 02FB C!
  6      54 C! 55 ! 6 #->
  7          11 CCMD C!  ( DRAW)
  8    0 BUFADR ! JSRCIO CKER ;
  9
 10  : PLOT >R OVER OVER OVER
 11    OVER DUP 0= IF 2+ ENDIF 1 -
 12    5A C! 5B ! I DRAWTO R>
 13    DRAWTO ;
 14   DECIMAL -->
 15

SCR # 16
  0  ( POS.     GR."  ) HEX
  1  : POS. 54 C! 55 ! ;
  2
  3  : GRTYPE -DUP IF OVER +
  4    SWAP DO I C@ 6 #-> 0B
  5    CCMD C! 0 BUFADR ! JSRCIO
  6    CKER DROP LOOP ELSE
  7     DROP ENDIF ;
  8  : GR(.") R COUNT DUP 1+ R>
  9    + >R GRTYPE ;
 10
 11  : GR." 22 STATE @ IF COMPILE
 12    GR(.") WORD HERE C@ 1+
 13    ALLOT ELSE WORD HERE COUNT
 14    GRTYPE ENDIF ; IMMEDIATE
 15        DECIMAL ;S

SCR # 19
  0 (ERROR MESSAGES )
  1 EMPTY STACK
  2 DICTIONARY FULL
```

```
  3 INCORRECT ADDRESS MODE
  4 NAME NOT UNIQUE
  5 LOCATE OUT OF RANGE
  6 DISK OUT OF RANGE
  7 FULL STACK
  8 DISK ERROR !!
  9 IN BOOT
 10
 11
 12
 13
 14
 15  QS FORTH VER 1.0 3/27/81

SCR # 20
  0 ( ERROR MESSAGES )
  1 COMPILATION ONLY, USE IN DEF
  2 EXECUTION ONLY
  3 CONDITIONALS NOT PAIRED
  4 INCOMPLETE DEFINITION
  5 IN PROTECTED DICTIONARY
  6 USE ONLY WHEN LOADING
  7 OFF CURRENT EDIT SCREEN
  8
  9 NOT COMPILED FROM DISK
 10 # OPERAND > $FF
 11 ILLEGAL USE OF Z-PAGE
 12 ILLEGAL ADDR MODE
 13
 14
 15

SCR # 21
  0  ( IOCB ERRORS )
  1  BREAK ABORT
  2  IOCB OPEN
  3  NONEXISTENT DEVICE
  4  IOCB WRITE ONLY
  5  INVALID COMMAND
  6  DEVICE NOT OPEN
  7  BAD IOCB #
  8  IOCB READ ONLY ERROR
  9  EOF
 10  TRUNCATED RECORD
 11  DEVICE TIMEOUT
 12  DEVICE NOT ACKNOWLEDGE CMD
 13  SERIAL BUS FRAMING ERROR
 14  CURSOR OUT OF RANGE
 15  SERIAL BUS FRAME OVERRUN

SCR # 22
  0  SERIAL CHECKSUM ERROR
  1  DEVICE ERROR
  2  BAD SCREEN MODE #
  3  FUNCTION NOT SUPPORTED
  4  SCREEN MODE EXCEEDED MEMORY
  5
  6
  7
```

```
 8
 9
10
11
12
13
14
15

SCR # 23
 0
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15

SCR # 24
 0  ( SET COLD START )
 1
 2  FORTH DEFINITIONS
 3
 4
 5  HEX
 6  ' FORTH 4 + @ C +ORIGIN !
 7  HERE  1E +ORIGIN !
 8  VOC-LINK @ 20 +ORIGIN !
 9  HERE     1C +ORIGIN ! ( FENCE)
10  HERE 600 - 80 / 2+
11  601 C!    ( BOOT CNT )
12  DECIMAL ;S
13
14
15

SCR # 25
 0 ( BOOTMAKER )
 1 HEX
 2     0 VARIABLE BOOTSTART
 3
 4 : MAKEBOOT
 5                 600 BOOTSTART !
 6    HERE BOOTSTART @ - 80 / 2+
 7           1 DO
 8    BOOTSTART @ I 0 R/W
 9         80 BOOTSTART +! LOOP ;
10
11
12
```

```
 13   DECIMAL ;S
 14
 15

SCR # 26
  0  ( MEMORY REF POINTER SET )
  1  HEX
  2
  3  : MEMSET
  4    02E5 @ ( FETCH HI LIMIT )
  5    DUP
  6    ' LIMIT !
  7    420 -
  8    DUP ' FIRST !
  9    DUP   USE  ! PREV  !
 10    COLD ;   ;S
 11
 12
 13
 14
 15

SCR # 30
  0  FORTH DEFINITIONS
  1  VOCABULARY EDITOR IMMEDIATE
  2  ( EDITOR CONT )    HEX
  3 : WHERE  ( PRINT ERROR )
  4     DUP B/SCR / DUP SCR !
  5     ." SCR # " DECIMAL . SWAP
  6     C/L /MOD C/L * ROT BLOCK +
  7     CR C/L TYPE CR HERE C@ -
  8     SPACES 5E EMIT ~[COMPILE]
  9     EDITOR QUIT ;
 10
 11 EDITOR DEFINITIONS
 12  CR ." EDITOR LOADING..."
 13
 14                     -->
 15

SCR # 31
  0  ( EDITOR CONT ) HEX
  1  : -MOVE ( BLOCK ADR->LINE )
  2      LINE C/L CMOVE UPDATE ;
  3
  4
  5  : E ( ERASE LINE )
  6    LINE C/L BLANKS UPDATE ;
  7
  8  : S DUP 1 - 0E DO I LINE I
  9     1+ -MOVE -1 +LOOP E ;
 10  : /R PAD 1+ SWAP -MOVE ;
 11                   -->
 12
 13
 14
 15

SCR # 32
```

```
 0  ( EDITOR CONT )
 1  : CLEAR SCR ! 10 0 DO FORTH
 2        I EDITOR E LOOP ;
 3
 4  : COPY B/SCR * OFFSET @ +
 5       SWAP B/SCR * B/SCR OVER
 6       + SWAP DO DUP FORTH
 7       I BLOCK 2 - ! 1+ UPDATE
 8       LOOP DROP FLUSH ;
 9
10  : D 0F DUP ROT DO I 1+ LINE
11      I -MOVE LOOP E ;
12
13
14  : P 1 TEXT /R ;
15            -->

SCR # 33
 0 ( DUPLICATE )
 1 0 VARIABLE EBLK  ( ENDING BLK )
 2 0 VARIABLE SBLK  ( STARTIN BLK )
 3 0 VARIABLE PSBLK
 4 : DISP ( ->DEST ADR IN FRE RAM )
 5    PSBLK @ B/BUF * HERE + ;
 6
 7 : GTPAR ( SET UP DO AND PSBLK )
 8    EBLK @ SBLK @ 0 PSBLK ! ;
 9
10 : MVIN ( MOVE BLKS INTO RAM )
11     GTPAR DO I BLOCK DISP
12     B/BUF CMOVE 1 PSBLK +!
13     LOOP ;
14          DECIMAL -->
15

SCR # 34
 0 : MOVOT ( WRITE RAM TO DISC )
 1     GTPAR OFFSET @ + SWAP OFFSET
 2     @ + SWAP  DO I BUFFER DISP
 3     SWAP B/BUF CMOVE 1 PSBLK +!
 4     UPDATE FLUSH LOOP ;
 5
 6  : DUPLICATE ( STARTSCR ENDSCR )
 7    1+ B/SCR * EBLK ! B/SCR *
 8     SBLK ! EBLK @ SBLK @ -
 9   ' FIRST 1+ C@ DP 1+ C@ -
10   2 * 2 -   > IF ." TOO MANY "
11    QUIT ENDIF CR    MVIN
12   ." INSERT DESTINATION DISK
13   " CR ." RETURN TO CONTINUE "
14   KEY DROP CR    MOVOT  ;
15        DECIMAL -->

SCR # 35
 0  ( ATARI FORTH EDITOR ) HEX
 1
 2  0 VARIABLE COL ( USR COL PTR )
 3  0 VARIABLE LIN ( USR LIN PTR )
 4
```

```
 5
 6  : EDLIST ( SPEC LIST FOR ED  )
 7    7D EMIT
 8    DECIMAL CR DUP SCR !
 9    ." SCR # " . 10 0 DO
10    CR I 3 .R I SCR @ .LINE
11    LOOP CR ;
12
13
14                        -->
15

SCR # 36
 0 ( ATARI ED OS ACCESS WORDS ) HEX
 1
 2  : ONCUR 0 02F0 C! ;
 3
 4  : OFCUR 1 02F0 C! ;
 5
 6   DECIMAL -->
 7
 8
 9
10
11
12
13
14
15

SCR # 37
 0  ( SMOVE )     DECIMAL
 1
 2  : SMOVE ( SOURCE DEST # TOMV )
 3     CR   FLUSH EMPTY-BUFFERS
 4    ." CAUTION !!! " CR
 5      >R 2DUP SWAP
 6   ." MOVE " DUP . ." THRU " R +
 7   1 - . ." -->" DUP . ." THRU "
 8   R + 1 - . SPACE ." Y OR N" CR
 9   R>   KEY 89 = IF
10   0 DO OVER I + OVER I +
11   COPY    LOOP DROP DROP
12   ELSE QUIT ENDIF ;
13  DECIMAL  -->
14
15

SCR # 38
 0  ( LFCUR RTCUR ) HEX
 1  : (LFCUR) 1E EMIT ;
 2  : (RTCUR) 1F EMIT ;
 3  DECIMAL
 4  : RTCUR OFCUR COL @ 31 =
 5        IF 31 0 DO (LFCUR) LOOP
 6        0 COL ! ELSE (RTCUR)
 7      1 COL +! ENDIF ONCUR ;
 8
 9
```

```
 10   : LFCUR OFCUR COL @
 11      IF (LFCUR) -1 COL +!
 12      ELSE 31 0 DO (RTCUR) LOOP
 13      31 COL ! ENDIF ONCUR ;
 14
 15  DECIMAL           -->

SCR # 39
  0 ( UPCUR DNCUR )  HEX
  1 : (DNCUR) 1D EMIT ;
  2 : (UPCUR) 1C EMIT ;
  3
  4 : DNCUR OFCUR LIN @
  5       F = IF F 0 DO (UPCUR)
  6              LOOP 0 LIN !
  7       ELSE (DNCUR) 1 LIN +!
  8       ENDIF ONCUR ;
  9
 10 : UPCUR OFCUR LIN @
 11    IF (UPCUR) -1 LIN +!
 12     ELSE F 0 DO (DNCUR)
 13     LOOP F LIN ! ENDIF
 14     ONCUR ;
 15  DECIMAL -->

SCR # 40
  0   ( HOME CURSOR ) HEX
  1
  2  : LINCLEAR ( CURSOR->LIN 0 )
  3     LIN @ DUP IF 0 DO UPCUR
  4     LOOP ELSE DROP ENDIF ;
  5  : COLCLEAR ( CURSOR->COL 0 )
  6     COL @ DUP IF 0 DO LFCUR
  7     LOOP ELSE DROP ENDIF ;
  8
  9  : HOMECUR ( CURSOR->HOME )
 10   LINCLEAR COLCLEAR ;
 11
 12
 13  : CURSHOW (RTCUR) (LFCUR) ;
 14
 15    -->

SCR # 41
  0 ( ED CONT EDCR...TAB ) DECIMAL
  1
  2  : BUFF-> ( BUFFER CHAR ADR )
  3           LIN @ SCR @ (LINE)
  4           DROP COL @ + ;
  5
  6  : EDCR ( SPECIAL CR FOR ED )
  7       COL @ IF COLCLEAR ENDIF
  8               DNCUR ;
  9
 10  5 VARIABLE (TAB)
 11  : TAB 31  COL @ - (TAB) @ <
 12       IF COLCLEAR ELSE
 13    (TAB) @ COL @ OVER MOD -
 14       0 DO RTCUR LOOP ENDIF ;
```

```
 15     DECIMAL -->

SCR # 42
  0  ( ED CONT EDMIT )   HEX
  1
  2  : ((EDEMIT))
  3      EMIT (LFCUR) RTCUR
  4      COL @ 0= IF DNCUR ENDIF ;
  5
  6  : TOBUFF ( SENDS CHAR TO LINE )
  7      DUP ( CHAR )
  8       BUFF-> C! ;
  9
 10  : EDMIT DUP 20 < IF BELL DROP
 11    ELSE TOBUFF ((EDEMIT))
 12    ENDIF  UPDATE 0 ;
 13       DECIMAL -->
 14
 15

SCR # 43
  0  ( LIN PRINT WORDS )  DECIMAL
  1  0 VARIABLE TEMP1
  2  0 VARIABLE TEMP2
  3
  4  : PTRSAV COL @ TEMP1 !
  5           LIN @ TEMP2 ! ;
  6
  7
  8  : LINOUT
  9        COLCLEAR BUFF-> 32
 10    TYPE  32 COL ! COLCLEAR ;
 11
 12   : CURREST  COLCLEAR TEMP1 @
 13     -DUP IF 0 DO RTCUR LOOP
 14          ENDIF ;
 15                        -->

SCR # 44
  0   HEX
  1
  2 : REFRESH ( OUTPUT ALL LINS)
  3                     PTRSAV
  4        10 LIN @ DO  LINOUT
  5           DNCUR LOOP TEMP2 @
  6       -DUP IF 0 DO DNCUR LOOP
  7            ENDIF  ;
  8
  9  DECIMAL -->
 10
 11
 12
 13
 14
 15

SCR # 45
  0  ( CHAR INSERT WORDS )
  1
```

```
 2  : MOVRT DUP OVER 1 - C@
 3         SWAP C! 1 - ;
 4
 5  : XPAND ( SPREAD LIN AT CUR )
 6      PTRSAV ( SAVE POINTERS )
 7             31  COL @ -
 8      DUP BUFF-> + SWAP 0
 9      DO MOVRT LOOP
10      BL SWAP C!
11      LINOUT CURREST UPDATE ;
12
13      DECIMAL -->
14
15

SCR # 46
 0  ( CHAR INSERT WORDS )
 1
 2  : MOVLF DUP OVER 1+  C@
 3         SWAP C! 1+  ;
 4
 5  : CPAND ( SHRINK LIN AT CUR )
 6      PTRSAV ( SAVE POINTERS )
 7      BUFF-> 31  COL @ -  0
 8      DO MOVLF LOOP
 9      BL SWAP C! ONCUR
10      LINOUT CURREST UPDATE ;
11
12      DECIMAL -->
13
14
15

SCR # 47
 0  HEX
 1
 2  : BKSP COL @ IF LFCUR ENDIF 20
 3      EDMIT LFCUR DROP ;
 4  : FINI ( WRAP-UP ON ESC )
 5      HOMECUR UPCUR (DNCUR)
 6      CR (UPCUR) ;
 7
 8  :  INSL ( SPREAD AT LIN # )
 9       LIN @ S REFRESH ;
10
11  : DELL ( DELETE LINE )
12       LIN @ D REFRESH ;
13               DECIMAL   -->
14
15

SCR # 48
 0  ( EDITOR LOOK UP TABLE ) HEX
 1  EDITOR DEFINITIONS
 2  10 VARIABLE  XTABLE
 3  1C C, ( UP ) 1D C, ( DN )
 4  1E C, ( LF ) 1F C, ( RT )
 5  7D C, ( HM ) 7E C, ( BS )
 6  0D C, ( CR ) 9D C, ( IL )
```

```
  7  9C C, ( DL ) FF C, ( XL )
  8  FE C, ( CL ) 7F C, ( TB )
  9  9F C, ( ST ) 9E C, ( CT )
 10  FD C, ( BL )  DECIMAL
 11  : KEYLIT  0 XTABLE @ 0
 12       DO DROP DUP I XTABLE 2
 13     + +   C@ = IF LEAVE ENDIF
 14             I LOOP ;
 15  DECIMAL  -->

SCR # 49
  0   ( CONTROL WORDS )
  1
  2  CASE: CONTROL
  3      UPCUR DNCUR LFCUR RTCUR
  4      HOMECUR BKSP  EDCR INSL
  5      DELL XPAND CPAND TAB
  6      BELL BELL  BELL EDMIT ;
  7
  8  : +KEY ( LIST BACK ONE )
  9      SCR @ DUP 1 >
 10      IF 1 - ENDIF EDLIST ;
 11
 12  : *KEY SCR @ 1+ EDLIST ;
 13
 14                     -->
 15

SCR # 50
  0  ( ED MODE CONTROL ) HEX
  1
  2 : ED (UPCUR) (RTCUR) (RTCUR)
  3   (RTCUR) (RTCUR)
  4    INVON
  5    1 COL ! F LIN ! HOMECUR
  6    CURSHOW BEGIN KEY
  7    DUP 1B XOR WHILE
  8    KEYLIT CONTROL  DROP
  9    CURSHOW REPEAT FINI
 10    INVOF
 11                  DROP ;
 12       DECIMAL  -->
 13
 14
 15

SCR # 51
  0  ( ED MODE CONTROL ) DECIMAL
  1 : 0-> DROP 0 ;
  2 : L        ( LIST SCREEN,WAIT )
  3     INVON        EDLIST
  4                  BEGIN KEY
  5   DUP 27 XOR WHILE
  6   DUP 43 = IF +KEY 0-> ENDIF
  7   DUP 42 = IF *KEY 0-> ENDIF
  8   DUP 45 = IF ED 0-> ENDIF
  9   IF BELL ENDIF REPEAT
 10   DROP  INVOF ;
 11
```

```
 12   FORTH DEFINITIONS
 13   : KL ~[COMPILE] EDITOR  EDITOR
 14     SCR @ L ; DECIMAL -->
 15

SCR # 52
  0  ( INVERSE ADDR, CNT ) HEX
  1   EDITOR DEFINITIONS
  2   : INTYPE INVON 0 DO I OVER +
  3       C@ 80 OR EMIT LOOP
  4        DROP INVOF ;
  5
  6   : <L> ( T IF BETWEEN L#'S )
  7       DUP SBLK @ < 0= SWAP
  8       EBLK @ > 0= AND ;
  9
 10   : IN.LINE  (LINE) INTYPE ;
 11
 12          DECIMAL -->
 13
 14
 15

SCR # 53
  0  ( ATARI FORTH EDITOR ) HEX
  1
  2  : INVLIST 7D EMIT
  3    DECIMAL CR DUP SCR !
  4    ." SCR # " . 10 0 DO
  5    CR I 3 .R I SCR @
  6    OVER <L> IF IN.LINE ELSE
  7    .LINE ENDIF  LOOP CR ;
  8
  9  0 VARIABLE STBLK
 10  0 VARIABLE SBBLK
 11  0 VARIABLE SSCR
 12  0 VARIABLE DTBLK
 13  0 VARIABLE DBBLK
 14  0 VARIABLE DSCR
 15         DECIMAL -->

SCR # 54
  0
  1  : FROM ( SCR LO HI L# )
  2    15 MIN
  3    DUP STBLK ! EBLK !
  4    DUP SBBLK ! SBLK ! DUP
  5    SSCR ! INVLIST ;
  6
  7  : /H LINE PAD 1+ C/L DUP PAD
  8        C! CMOVE ;
  9
 10  : SS->DD
 11    STBLK @ 1+ SBBLK @ - 0
 12    DO SSCR @ SCR ! I SBBLK @ +
 13   /H  DSCR @ SCR ! I DBBLK @ +
 14   /R  LOOP ;
 15               DECIMAL -->
```

```
SCR # 55
  0
  1
  2
  3
  4  : INTO DUP SBLK ! DBBLK !
  5    DSCR ! STBLK @ SBBLK @ -
  6    DBBLK @ + DUP EBLK ! DTBLK !
  7     SS->DD   DSCR @ INVLIST
  8     CR ." OK? Y/N " KEY
  9     89 = IF KL ELSE
 10     EMPTY-BUFFERS ENDIF ;
 11
 12
 13         ;S
 14
 15

SCR # 58
  0  ( ATARI ASSMBLER  9/19/80 )
  1  FORTH DEFINITIONS
  2  VOCABULARY ASSEMBLER IMMEDIATE
  3  ' ASSEMBLER CFA
  4  ' ;CODE  8 + !
  5  10 VARIABLE ADRMD
  6  : CODE: ?EXEC    !CSP
  7      10 ADRMD !    CREATE
  8     ~[COMPILE] ASSEMBLER  ;
  9    IMMEDIATE
 10
 11  : C; CURRENT @ CONTEXT !
 12       ?EXEC ?CSP SMUDGE ;
 13    IMMEDIATE
 14  CR ." ASSEMBLER LOADING..."
 15                      -->

SCR # 59
  0  ( MSC LABELS TO FIG CODE )
  1
  2  ASSEMBLER DEFINITIONS
  3
  4  HEX
  5  47  +ORIGIN CONSTANT NEXT
  6  3DF +ORIGIN CONSTANT PUSH0A
  7  B5  CONSTANT XSAVE
  8
  9  0 VARIABLE INCLS
 10  DECIMAL -->
 11
 12
 13
 14
 15

SCR # 60
  0  -->
  1
  2
  3
```

```
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15

SCR # 61
 0  ( OPCODE TABLE )   HEX
 1  0 VARIABLE OPTBL ( FF ILLEGAL )
 2 ( A:)   FF C, FF C, FF C, FF C,
 3         08 C, FF C, FF C, FF C,
 4 ( 16)   0C C, 08 C, 08 C, 08 C,
 5         0C C, 0C C, 0C C, 0C C,
 6 ( 8A)   04 C, 00 C, 00 C, 00 C,
 7         04 C, 04 C, 04 C, 04 C,
 8 ( # )   08 C, FF C, FF C, FF C,
 9         FF C, 00 C, 00 C, 00 C,
10 ( 16,X) 1C C, 18 C, FF C, FF C,
11         1C C, FF C, FF C, 1C C,
12 ( 16,Y) 18 C, FF C, FF C, FF C,
13         FF C, FF C, 1C C, FF C,
14
15               DECIMAL   -->

SCR # 62
 0 ( '8,X' )      HEX
 1      00 C, FF C, FF C, FF C,
 2      FF C, FF C, FF C, FF C,
 3 ( '8',Y)
 4      10 C, FF C, FF C, FF C,
 5      FF C, FF C, FF C, FF C,
 6 ( 8,X )
 7      14 C, 10 C, FF C, 10 C,
 8      14 C, FF C, FF C, 14 C,
 9 ( 8,Y )
10      FF C, FF C, 10 C, FF C,
11      FF C, FF C, 14 C, FF C,
12
13
14
15           DECIMAL   -->

SCR # 63
 0  ( TABLE FETCH WORDS )   DECIMAL
 1
 2  : ?TABLE
 3      ADRMD @ 8 *
 4      INCLS @ +
 5      OPTBL 2 + + C@ ;
 6
 7  HEX
 8  : AMDCK  ?TABLE
```

```
 9       DUP FF = 1C ?ERROR ;
10  ( ADR MODE ERROR )
11                    DECIMAL -->
12
13
14
15

SCR # 64
 0   HEX
 1  : ?HI DUP FF00 AND ;
 2
 3  DECIMAL
 4  : ADRMD! ADRMD ! ;
 5
 6  : 16/8     ?HI
 7       IF   ( LONG  ADR )
 8           1
 9       ELSE ( SHORT ADR )
10           2 ENDIF ADRMD ! ;
11
12  : #: ?HI 26 ?ERROR
13                3 ADRMD! ;
14
15   : A: 0 ADRMD! ; DECIMAL   -->

SCR # 65
 0
 1  : ,X  ?HI ( TEST FOR 16/8 )
 2            IF  ( 16 )
 3               4
 4            ELSE  8
 5            ENDIF ADRMD! ;
 6
 7
 8
 9                    -->
10
11
12
13
14
15

SCR # 66
 0
 1  : ,Y ?HI
 2           IF ( 16 OR 8 )
 3               5
 4                 ELSE
 5               9
 6           ENDIF ADRMD! ;
 7
 8
 9
10                    -->
11
12
13
```

```
 14
 15

SCR # 67
  0  : ,X)
  1       ?HI  27 ?ERROR
  2              6 ADRMD! ;
  3
  4  : ),Y ?HI 27 ?ERROR
  5              7 ADRMD! ;
  6
  7                     -->
  8
  9
 10  ( 1B IS Z-PAGE ERROR MSG )
 11
 12
 13
 14
 15

SCR # 68
  0    ( CODE BUILDERS AAA CLASS )
  1
  2  : BLDCD INCLS ! ( CLASS )
  3        10 ADRMD @ =
  4    IF SWAP 16/8 SWAP  ENDIF
  5       AMDCK OR C, ( TEST MODE )
  6       ADRMD @ IF  ( NOT A: )
  7       ?HI IF , ELSE C, ENDIF
  8             ENDIF 10 ADRMD ! ;
  9
 10   : T3A <BUILDS SWAP  C, C,
 11           DOES>
 12           DUP C@ SWAP 1+ C@
 13           BLDCD ;
 14      -->
 15

SCR # 69
  0  ( OPCODE FOLLIES ) HEX
  1  ( AAA CLASS INCLS 0 )
  2
  3  61 0 T3A ADC, 21 0 T3A AND,
  4  C1 0 T3A CMP, A1 0 T3A LDA,
  5  01 0 T3A ORA, E1 0 T3A SBC,
  6  81 0 T3A STA, 41 0 T3A EOR,
  7
  8  ( BB1 CLASS INCLS 1 )
  9
 10  C6 1 T3A DEC, E6 1 T3A INC,
 11
 12  ( BBX CLASS INCLS 2 )
 13
 14  86 2 T3A STX,
 15              DECIMAL -->

SCR # 70
  0  ( BBY CLASS 3 INCLS )   HEX
```

```
 1
 2  84 3 T3A STY,
 3
 4  ( BBB CLASS 4 INCLS )
 5
 6  02 4 T3A ASL,  42 4 T3A LSR,
 7  22 4 T3A ROL,  62 4 T3A ROR,
 8
 9  ( CC CLASS 5 INCLS )
10
11  E0 5 T3A CPX,  C0 5 T3A CPY,
12
13
14
15                DECIMAL -->

SCR # 71
 0  ( DDDX CLASS 6 INCLS )  HEX
 1
 2  A2 6 T3A LDX,
 3
 4  ( DDDY CLASS 7 INCLS )
 5
 6  A0 7 T3A LDY,
 7
 8
 9
10
11
12
13                DECIMAL -->
14
15

SCR # 72
 0  : IMPL <BUILDS C, ( IMPLIED )
 1      DOES> C@ C,       ; HEX
 2
 3   00 IMPL BRK, 18 IMPL CLC,
 4   D8 IMPL CLD, 58 IMPL CLI,
 5   B8 IMPL CLV, CA IMPL DEX,
 6   88 IMPL DEY, E8 IMPL INX,
 7   C8 IMPL INY, EA IMPL NOP,
 8   48 IMPL PHA, 8A IMPL TXA,
 9   98 IMPL TYA, 08 IMPL PHP,
10   68 IMPL PLA, 28 IMPL PLP,
11   40 IMPL RTI, 60 IMPL RTS,
12   38 IMPL SEC, F8 IMPL SED,
13   78 IMPL SEI, AA IMPL TAX,
14   A8 IMPL TAY, BA IMPL TSX,
15   9A IMPL TXS,  DECIMAL -->

SCR # 73
 0  ( REL BRANCH )     DECIMAL
 1
 2  HEX
 3  : RBR <BUILDS C,
 4          DOES> C@ C, 3 C, ;
 5
```

```
 6  ( BRANCH AROUND JMP )
 7
 8  90 RBR BCC, B0 RBR BCS,
 9  F0 RBR BEQ, 30 RBR BMI,
10  D0 RBR BNE, 10 RBR BPL,
11  50 RBR BVC, 70 RBR BVS,
12                DECIMAL -->
13
14
15

SCR # 74
 0  ( JMP & BIT )      HEX
 1
 2  : JMP, 4C C, , ;
 3
 4  : (JMP), 6C C, , ;
 5
 6  : JSR, 20 C, , ;
 7
 8  : BIT, ?HI  IF 2C C, ,
 9              ELSE 24 C, C,
10         ENDIF   ;
11
12
13
14
15                DECIMAL   -->
```